# Development and Analysis of Stochastic Boundary Coverage Strategies for Multi-Robot Systems

by

Ganesh P Kumar

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved March 2016 by the
Graduate Supervisory Committee:

Spring Berman, Co-Chair
Georgios Fainekos, Co-Chair
Rida Bazzi
Violet Syrotiuk
Thomas Taylor

ARIZONA STATE UNIVERSITY

May 2016

ABSTRACT

Robotic technology is advancing to the point where it will soon be feasible to deploy massive populations, or swarms, of low-cost autonomous robots to collectively perform tasks over large domains and time scales. Many of these tasks will require the robots to allocate themselves around the boundaries of regions or features of interest and achieve target objectives that derive from their resulting spatial configurations, such as forming a connected communication network or acquiring sensor data around the entire boundary. We refer to this spatial allocation problem as boundary coverage. Possible swarm tasks that will involve boundary coverage include cooperative load manipulation for applications in construction, manufacturing, and disaster response.

In this work, I address the challenges of controlling a swarm of resource-constrained robots to achieve boundary coverage, which I refer to as the problem of stochastic boundary coverage. I first examined an instance of this behavior in the biological phenomenon of group food retrieval by desert ants, and developed a hybrid dynamical system model of this process from experimental data. Subsequently, with the aid of collaborators, I used a continuum abstraction of swarm population dynamics, adapted from a modeling framework used in chemical kinetics, to derive stochastic robot control policies that drive a swarm to target steady-state allocations around multiple boundaries in a way that is robust to environmental variations.

Next, I determined the statistical properties of the random graph that is formed by a group of robots, each with the same capabilities, that have attached to a boundary at random

locations. I also computed the probability density functions (pdfs) of the robot positions and inter-robot distances for this case.

I then extended this analysis to cases in which the robots have heterogeneous communication/sensing radii and attach to a boundary according to non-uniform, non-identical pdfs. I proved that these more general coverage strategies generate random graphs whose probability of connectivity is Sharp-P Hard to compute. Finally, I investigated possible approaches to validating our boundary coverage strategies in multi-robot simulations with realistic Wi-fi communication.

*To Apu, Appa, Amma, and Thatha*

# ACKNOWLEDGMENTS

My wife Apeksha was a tremendous help and source of support for the past three years; the best decision I took during the program was to get married to her. As were my parents, grandfather, and my cousins Aroon and Kavita, who stood by my side throughout these 5.5 years. These years also saw my maternal grandpa and paternal grandma passing away; to my great sorrow, they are not here to witness the culmination of my post-tertiary education.

A big word of thanks goes to my advisor, Prof. Berman, for her general guidance and her patient proofreading and editing throughout the process. I had many fruitful discussions with my committee members on complexity theory (Prof.Bazzi), Wi-fi communication (Prof. Syrotiuk), and probabilistic computations (Prof. Taylor). Prof. Ted Pavlic was a big source of ideas throughout my work with Prof. Berman. I also acknowledge the help of my labmates, Sean and Ruben (for hardware-related discussions, and Sean being a continual source of recreation), Karthik, Ragesh, Hamed, Shiba, Chase, Yuan, and Jared. I enjoyed teaching them and answering their questions about CS and software engineering in general, and am happy to have been a source of practical help. As a result of all this discussion, I have a good idea of how to teach Python, C++ , Arduino programming, and OpenCV to undergraduates!

A number of my friends were a big source of moral support: ex-labmates Yucong and Benjamin; Dr. Buffin and Prof. Pratt; IIIT friends Mayur, Bhavani, Rajnish; my ex-Motorola colleagues, especially Amruth, Nayan, and Wajahat; others Ph.D. students at ASU: Hemanth, Prasanth, and Yoalli. My ex-team at Mayfield/Bosch also contributed significantly to my knowledge, particularly my supervisors Dr. Kaijen Hsiao and Dr. Sarah Osentoski.

TABLE OF CONTENTS

Page

LIST OF TABLES ..... xi

LIST OF FIGURES ..... xii

CHAPTER

1 INTRODUCTION ..... 1

1.1 State of the Art in Robotic Swarms ..... 1

1.2 Defining Multi-Robot Collective Transport and Boundary Coverage ..... 3

1.3 Motivation ..... 4

1.3.1 Assumptions about Robot Capabilities ..... 4

1.3.2 Problem Statement ..... 6

1.4 Novel Contributions ..... 8

1.5 Outline of Dissertation ..... 9

1.5.1 Part 1 : Ant-based CT ..... 9

1.5.2 Part 2 : Multi-boundary Coverage ..... 10

1.5.3 Parts 3 and 4: Maintaining a Wireless Network around a Boundary 11

1.5.4 Part 5 : Experimental Investigation of BC with Wireless Communications ..... 12

2 LITERATURE REVIEW ..... 13

2.1 Collective Transport in Robotic Swarms (Wilson *et al.*, 2014) ..... 13

2.2 Collective Transport in Ants (Kumar *et al.*, 2013) ..... 14

2.3 Multi-Robot Boundary Coverage and Task Allocation (Pavlic *et al.*, 2013a; Kumar and Berman, 2014) ..... 16

2.4 Maintaining a Wireless Network on the Boundary ..... 17

2.4.1 Wireless Networks as Point Processes ..... 17

2.4.2 Computational Geometry ..... 18

CHAPTER Page


2.5 Multi-robot and Wi-fi Simulation . . . . . 18
2.5.1 ROS . . . . . 19
2.5.2 ARGoS . . . . . 20
2.5.3 NS3 . . . . . 20
2.5.4 Pylayers . . . . . 21
2.5.5 GENI . . . . . 21
3 BACKGROUND . . . . . 22
3.1 Chemical Reaction Networks (CRNs) . . . . . 22
3.2 Stochastic Hybrid Systems . . . . . 23
3.3 Computational Geometry . . . . . 25
3.4 Order Statistics . . . . . 26
3.4.1 Probability Integral Transform . . . . . 27
3.4.2 Order Statistics for Non-IID Parents . . . . . 28
3.5 Counting Complexity . . . . . 28
3.5.1 Computing Permanents is #PH . . . . . 29
3.6 Generalized Simplex Hypercube Intersection . . . . . 30
3.6.1 Recasting VHSP as a Simplex Hypercuboid Intersection . . . . . 30
3.7 Poisson Point Processes (PPP) and Random Geometric Graphs (RGG) . . 32
3.7.1 Random Geometric Graphs . . . . . 33
3.7.2 PPPs in $\mathscr{B}$ . . . . . 34
3.7.3 Matern Hard Core Processes in $\mathscr{B}$ . . . . . 34
4 COLLECTIVE TRANSPORT IN THE DESERT ANT *A. cockerelli* . . . . . 36
4.1 Abstract . . . . . 36
4.2 Experimental Trials . . . . . 37

CHAPTER Page

4.3 Stochastic Hybrid System Model . . . . . 40
4.4 Computing Moments of the State Vector . . . . . 43
4.5 Estimating SHS Parameters . . . . . 44
4.5.1 Note on Model Fitting Process . . . . . 45
4.6 Inferences from Estimated Parameters . . . . . 46
5 STOCHASTIC CONTROL POLICIES FOR THE COVERAGE OF MULTIPLE LOAD BOUNDARIES . . . . . 49
5.1 Problem Statement . . . . . 50
5.2 Design of Stochastic Controller . . . . . 51
5.3 Microscopic Model: Chemical Reaction Network . . . . . 52
5.4 Macroscopic Model: Concentration Fields . . . . . 53
5.5 Stability of the Nontrivial Equilibrium . . . . . 55
5.5.1 Asymptotic Stability of the Nontrivial Equilibrium . . . . . 56
5.6 Reduced-Order Model and its Applications . . . . . 56
5.6.1 Convergence Rate and Variance . . . . . 57
5.6.2 Optimal Choice of Control Policy . . . . . 58
5.7 Correction for Spatial Effects . . . . . 60
5.8 Control Synthesis for Multiple Disk Types . . . . . 60
5.9 Validating the Stochastic Controller . . . . . 62
5.10 Simulating Ant-based Collective Transport . . . . . 62
5.11 Comparison with Existing Work . . . . . 65
5.11.1 Comparison with (Konur *et al.*, 2012) . . . . . 65
5.11.2 Comparison with Amoebot . . . . . 68
5.12 Conclusion . . . . . 70

CHAPTER Page

6 MAINTAINING A WIRELESS NETWORK AROUND A BOUNDARY - PART I . . . . . 71
6.1 Overview and Problem Statement . . . . . 71
6.1.1 Models of Boundary Coverage . . . . . 73
6.1.2 Problem Statement . . . . . 76
6.1.3 Summary of Results . . . . . 77
6.2 Deterministic Coverage Strategy (DCS) for $\mathscr{B}$ . . . . . 77
6.2.1 Deciding Connectivity of $\mathscr{G}$ . . . . . 80
6.2.2 Maintaining a List of Robot Positions . . . . . 80
6.3 Slack and Position Simplices . . . . . 81
6.4 Computing the Probability of Connectivity . . . . . 83
6.4.1 Decomposing $\mathscr{U}$ into Simplices . . . . . 86
6.4.2 Alternative Approach to Computing $\mathrm{Vol}(\mathscr{F})$ . . . . . 87
6.4.3 Monte Carlo Method for Computing $p_{\mathsf{con}}$ . . . . . 88
6.5 Computing Densities of Positions and Slacks . . . . . 89
6.5.1 Marginal PDF of $\mathtt{X}_i$ . . . . . 90
6.5.2 Marginal PDF of $S_i$ . . . . . 91
6.6 Shape of $\mathscr{F}$ . . . . . 92
6.6.1 Vertices of $\mathscr{F}$ . . . . . 93
6.7 Marginal PDFs over $\mathscr{F}$ . . . . . 97
6.7.1 Small $n$ Case . . . . . 97
6.7.2 Large $n$ Case . . . . . 98
6.7.3 REPSAMP: Fast Random Sampling of $\mathscr{F}$ . . . . . 99
6.8 Validation . . . . . 101

CHAPTER Page

6.8.1 Probability of Connectivity, $p_{\mathsf{con}}$ . . . . . 102
6.8.2 Marginal PDFs of $\mathtt{X}_i$ and $S_i$ . . . . . 102
6.8.3 Covariance Matrices of Positions and Slacks . . . . . 103
6.8.4 Statistics at Connectivity: Small $n$ . . . . . 104
6.8.5 Statistics at Connectivity: Large $n$ . . . . . 106
6.8.6 Equivalence of Algorithms 2 and 4 . . . . . 108
7 MAINTAINING A WIRELESS NETWORK AROUND A BOUNDARY - PART II . . . . . 110
7.1 Summary of Results . . . . . 110
7.2 IID Coverage by Point Robots . . . . . 110
7.2.1 Joint PDF of Positions . . . . . 111
7.3 Properties of $\mathscr{G}$ generated by IID Parents . . . . . 112
7.3.1 Probability of Connectivity . . . . . 112
7.3.2 Number of Connected Components . . . . . 113
7.3.3 Coverage Length . . . . . 114
7.3.4 Edge Count . . . . . 114
7.3.5 Dynamic Coverage with IID Attachment . . . . . 115
7.4 Properties of $\mathscr{G}$ Computed by Monte-Carlo Sampling . . . . . 118
7.4.1 Uniform parent . . . . . 118
7.4.2 Piecewise Uniform Parent . . . . . 118
7.5 Uniform Coverage by Finite Robots . . . . . 121
7.5.1 Connectivity of Conflict-Free Parking . . . . . 124
7.6 Computing $p_{\mathsf{con}}$ for Uniform Coverage . . . . . 126
7.6.1 Computing $Vol(\mathscr{F})$ and Vol$(\mathscr{U})$ for Uniform Parents . . . . . 127

CHAPTER Page

7.7 PCON for Nonuniform IID parents . . . . . 128
7.7.1 More General Parents whose PCON is #PH . . . . . 130
7.8 PCON for Robots with Distinct Communication Ranges . . . . . 131
7.9 PCON for INID Parents . . . . . 137
8 FUTURE SIMULATIONS . . . . . 139
8.1 Design of Integrated Simulator . . . . . 139
8.1.1 Execution of Boundary Coverage using Algorithm 1 . . . . . 140
8.2 Current Challenges to Implementation . . . . . 140
8.2.1 Steps to get a Working Simulation . . . . . 141
9 CONCLUSION AND FUTURE WORK . . . . . 142
9.1 Part 1 : Ant-Based CT . . . . . 142
9.2 Part 2 : Multi-Boundary Coverage . . . . . 143
9.3 Parts 3 and 4: Maintaining a Wireless Network across a Boundary . . . . . 145
9.4 Part 5: Simulating Boundary Coverage . . . . . 147
REFERENCES . . . . . 148
APPENDIX
A LINKS TO VIDEO AND CODE RESOURCES . . . . . 156
A.1 Videos . . . . . 157
A.2 Code Repositories . . . . . 157
B STATEMENT ON PERMISSION FROM CO-AUTHORS . . . . . 158

LIST OF TABLES


Table Page

3.1 Complexity Classes and Problems . . . . . 32

4.1 Standard Error in WLS Fitting . . . . . 46

5.1 Comparison between Robozyme and Foraging Models . . . . . 67

5.2 Comparison between Robozyme and Amoebot Models . . . . . 69

6.1 Typographic Conventions . . . . . 72

6.2 Point Robots: $p_{\mathsf{con}}$ vs. $\hat{p}_{\mathsf{con}}$ . . . . . 103

6.3 Estimating $\mathtt{X}$ and $\mathbf{S}$ . . . . . 103

6.4 Estimating Position Covariance Matrices . . . . . 104

6.5 Estimating Slack Covariance Matrices . . . . . 104

6.6 Estimating $\mathbf{S}|\mathsf{Con}$ and $\mathtt{X}|\mathsf{Con}$ for small $n$ . . . . . 105

6.7 Estimating Slack Covariance Matrices for Small $n$ . . . . . 105

6.8 Equivalence of Sampling Algorithms - 1 . . . . . 108

6.9 Equivalence of Sampling Algorithms - 2 . . . . . 109

LIST OF FIGURES


Figure Page

4.1 *Aphaenogaster cockerelli* ants Transporting Artificial Load . . . . . 38

4.2 Trajectory of Artificial Load . . . . . 39

4.3 Variables $N_F$ and $v_L$ are Uncorrelated . . . . . 40

4.4 Observed vs Predicted State Variables for 3 Trials . . . . . 47

4.5 Results of WLS Fitting of State Variables . . . . . 48

5.1 Example Scenario with 2 Disk Types . . . . . 50

5.2 Control Flow of Stochastic Control Policy . . . . . 52

5.3 Phase Plot of $r$ . . . . . 57

5.4 Effect of Varying Disk Mixture . . . . . 59

5.5 Effect of Varying Environmental Parameters . . . . . 63

5.6 Simulated Trajectories of Robot Allocation . . . . . 64

6.1 Configuration of Point Robots . . . . . 82

6.2 Position and Slack Simplices for $n = 2$ Robots. . . . . 83

6.3 Simplex $\mathscr{S}$ and Hypercube $\mathscr{H}$ for $n = 1$ . . . . . 85

6.4 Connected Region $\mathscr{F}$ for Small $n$ . . . . . 92

6.5 Connected Region $\mathscr{F}$ for Large $n$ . . . . . 92

6.6 Connected Region $\mathscr{F}$ in $\mathscr{S}$ for $n = 2$. . . . . 95

6.7 Hexagonal $\mathscr{F}$ for $n = 2$ and $d > s/2$ . . . . . 96

6.8 3-D Plot of $p_{\mathsf{con}}$ . . . . . 102

6.9 Beta PDFs Fit to Frequency Plot of Connected $\mathtt{X}_2$ . . . . . 106

6.10 Beta PDF fit to Frequency Plot of Connected $S_2$ . . . . . 108

7.1 Expected Properties of $\mathscr{G}$ Generated by Uniform Parent - 1 . . . . . 119

7.2 Expected Properties of $\mathscr{G}$ Generated by Uniform Parent - 2 . . . . . 120

7.3 Expected Properties of $\mathscr{G}$ Generated by 3-PWU Parent - 1 . . . . . 122

Figure Page

7.4 Properties of $\mathscr{G}$ Generated by 3-PWU parent - 2 . . . . . 123

7.5 Piecewise Uniform PDF of 4 Pieces . . . . . 129

7.6 1-Hop-Connected Configuration of Point Robots with Distinct Ranges . . . . . 132

7.7 Robots with Distinct Noise Profiles have INID Parents . . . . . 138

# Chapter 1

# INTRODUCTION

## 1.1 State of the Art in Robotic Swarms

Robotic swarms have the potential to collectively perform tasks over very large domains and time scales, succeeding even in the presence of failures, errors, and disturbances (Brambilla *et al.*, 2013). Low-cost miniature autonomous robots for swarm robotic applications are currently being developed as a result of recent advances in computing, sensing, actuation, power, control, and manufacturing technologies (Mehta *et al.*, 2015; Wood *et al.*, 2013; Kitts and Egerstedt, 2008). In addition, micro and nanoscale platforms such as DNA machines, synthetic bacteria, magnetic materials, and nanoparticles are being designed for biomedical and manufacturing applications (Mavroidis and Ferreira, 2013; Diller and Sitti, 2013), in which they would need to be deployed in massive numbers. The past few years have seen an increasing amount of investment in home robots, drones, and robotic vehicles (Alterovitz *et al.*, 2014), by numerous startups such as Uber, Google X, and Mayfield Robotics (Westerman and Bonnet, 2015). We present some of these applications, paraphrasing from (Georgia Institute of Technology, et.al., 2013) and (US Department of Defense, 2013).

1. *Manufacturing*: This sector represents 14% of the United States' GDP and 70% of its net export. The past half-century has seen a tremendous rise in the use of robots in manufacturing, especially in automotives and electronics. While automotive manu-

facturers such as GM, Boeing and Lockheed Martin have traditionally used industrial robots for aircraft assembly, the electronics industry has recently seen a rise in the employment of robots, with Apple, Lenovo, and Samsung using robot teams for various hardware manufacturing tasks.

2. *Defense*: The US Department of Defense (DoD) employs a large number of robotic vehicles, in land, water, and air. The low-cost autonomous attack system (LOCAAS) is an aerial robotic swarm developed by the DoD to autonomously search for and destroy enemy targets.

3. *Service Applications*: Robots are now finding a presence in professional and domestic service applications. Autonomous lawnmowers and vacuum cleaners (such iRobot Roomba and Create (Mataric *et al.*, 2007)) are the two most familiar examples of their use in domestic service. Professionally, robots are being used for power plant and infrastructure inspection, as well as in logistics such as the delivery of bedding and pharmaceuticals at hospitals.

4. *Space*: The Spirit, Opportunity, and Curiosity Mars rovers are exemplars for the use of robots in space applications. Further, the Robonaut swarm developed by NASA and GM for use in the International Space Station is one example of the use of swarms in space exploration (Bekey *et al.*, 2006).

## 1.2 Defining Multi-Robot Collective Transport and Boundary Coverage

The objective of this dissertation is to summarize all our work done so far towards solving the problem of *Boundary Coverage (BC)* with a multi-robot team. We define the problem of multi-robot *collective transport (CT)* as one in which a team of autonomous agents is required to transport a heavy payload from a starting point to a destination. The destination may either be prescribed or be a location chosen consensually by the robot team. CT is a necessary capability for diverse applications of autonomous multirobot systems including construction, warehousing, manufacturing, and search-and-rescue operations (Parker, 2015). Possible instances of collective transport include transporting wreckage and people away from a disaster scene (Zhang *et al.*, 2013), carrying heavy materials around a construction site (Guizzo, 2008) , and conveying merchandise from one part of a warehouse to another. In many of these problem scenarios, the agents transporting the payload would lack prior knowledge of the load and the environment; further, they may have unreliable information about their positions with respect to a global coordinate system. Moreover, these agents would be capable of sensing the environment, and communicating with each other, only over a limited range. These constraints make it challenging to develop a comprehensive solution to collective transport.

Boundary coverage (BC) is the first nontrivial subproblem in the process of solving CT. We define a *Boundary Coverage Scheme (BCS)* to be a process by which a multi-robot team autonomously arranges itself around the boundary of an object or region of interest. BC has wide applications, including environmental monitoring, exploration, disaster response, surveillance, perimeter patrolling, and even nanomedicine, where micro and nano robotic

swarms may be used for drug delivery and diagnostics (Tong *et al.*, 2013; Hauert *et al.*, 2013). For example, we may require ligands and antibodies to bind to a polymeric drug-carrying nanoparticle in a specific way to aid drug delivery to tumor cells (Sinha *et al.*, 2006). The coming chapters will have a particular focus on schemes in which each robot in the team attaches to a random position on the boundary.

## 1.3 Motivation

The swarm robotic platforms of Section 1.1 are subject to certain limitations that restrict the types of control strategies that they can execute. As stated earlier, each robotic platform has limited onboard power, computation, sensing, and communication capabilities, and may lack global localization. The robot platforms exhibit *stochastic* behaviors that arise from noise due to sensor and actuator errors, randomness in their encounters with boundaries, and at the nanoscale, the effects of Brownian motion and chemical interactions. These sources of stochasticity produce uncertainty in the locations of eventual robot encounters with a boundary. For this reason, we refer to the task of BC executed by the robotic platforms as *stochastic boundary coverage*, and define a *stochastic coverage schemes (SCS)* to be one that exploits stochasticity.

### *1.3.1 Assumptions about Robot Capabilities*

We make the following assumptions about the capabilities of the multi-robot team. We assume that each robot can locally sense its environment and communicate with other robots nearby, within a communication radius. Each robot can identify fellow robots (for instance

by their color), and can also identify the load boundary. As stated earlier, robots lack global localization, but can execute a programmed random motion. Consequently, in the tasks that we consider, robots must allocate themselves around the boundary of a region or object according to a *target probability density function (pdf)*. Further, we also assume that the robots have sufficient memory to store the length of the boundary, the locations of nearby robots, and distances to them, a fact that will be used in the SCS in Chapter 6 and Chapter 7. We will generally assume that no robot crashes or otherwise fails while executing its coverage scheme.

We are specifically interested in SCS for three reasons. Firstly, stochastic strategies may be simpler to implement than their deterministic counterparts on robots with minimal capabilities. In contrast to SCS, *deterministic coverage schemes (DCS)* may require more capabilities such as precise navigation and localization than most swarm platforms support. Secondly, SCS may be more robust to the type of the boundary they cover, and may avoid deadlocks more effectively than deterministic algorithms, as evinced by collective transport in *Aphaenogaster cockerelli*.

Finally, from a computational perspective, they provide valuable insights into the *average* positions and inter-robot distances generated by multi-robot BC. Here, "average" is not uniquely defined, but rather depends on the underlying spatial pdf of attachment. This underlying spatial pdf is dictated by the specific problem we have to solve, for example:

1. In a scenario in which the robot team is required to carry wreckage away from a disaster scene, it may be desirable to allot more (resp. less) robots to locations with a higher (resp. lower) density of wreckage. A solution requires programming each

robot with a correspondingly *nonuniform* spatial pdf around the perimeter of wreckage.

2. Consider the adsorption of several types of particles on a substrate. Each type of particle has a different adsorption profile, so that distinct types of particles have non-identical spatial pdfs on the substrate.

This computational perspective will be exploited in Chapter 6 and Chapter 7 to determine the probability that the communication graph of robots that communicate within a certain radius will be connected. Thus, if we need the boundary to be covered by a communication network of robots with a desired probability, our computations will tell us the number of robots to allot to the coverage task, and the minimum communication radius each of them needs to have. In short, our results about the average case relate the the target coverage statistics to designable system parameters such as robot dimensions and population size. Further, these computations may be used to design and evaluate distributed data structures that maintain network connectivity in the event of failures, and when attached robots need to reallocate to other tasks.

This work addresses five different problems on BC.

### *1.3.2 Problem Statement*

**Part** 1 Develop analytical models of group retrieval behaviors observed in ants colonies. Draw inferences about mechanisms that underlie robust group behaviors from these models, and apply them to controller design for multi-robot CT.

**Part** 2 Consider a scenario in which a swarm of robots must allocate among a collection of loads that are distributed randomly in an environment. Assume that robots have no global localization or communication capabilities. Develop a strategy whereby they can attach to the load boundaries in a way that guarantees a desired allocation ratio.

**Part** 3 Consider a multi-robot team that attaches to a load boundary, with uniform iid (independent and identically distributed) probability density function (pdf) over the boundary. Suppose they need to maintain a communication network over the boundary, and thereby monitor the boundary. Assuming each has a known radius of communication, compute the resulting joint pdf of robot positions. Further, compute the probability that the communication graph formed by the swarm is connected, the distance between nearest neighbors, and the longest uncovered distance.

**Part** 4 Extend this analysis from Part 3 above by allowing robots to have distinct communication radii, non-uniform iid pdfs, and inid (independent and non-identically distributed) spatial pdfs over the boundary. Determine the computational complexity of determining the relevant probability of connectedness, using complexity-theoretic reductions, to determine what variants of the problem are easy and what are hard.

**Part** 5 Investigate protocols for multi-robot BC that use Wi-fi communication, using either simulation or hardware-based experiments. Elucidate the role of communication in BC, and detail what happens when communication breaks down.

## 1.4 Novel Contributions

We have developed a novel framework for the analysis and design of stochastic robot control policies that achieve multi-robot boundary coverage. Our framework: (1) employs a*dynamical models* of the tasks involved in the coverage process, in a way that facilitates its theoretical analysis; (2) permits robots to *dynamically join and leave* a boundary; and (c) is *decentralized*, i.e. it does not require a centralized coordinator for system operation. My collaborators and I were the first to accomplish the following, with results published in the references indicated:

1. Develop a hybrid dynamical system model of collective transport in ants from experimental data ( *HSCC 2013* (Kumar *et al.*, 2013)) and use it to define robot control policies for ant-like transport behaviors (*Swarm Intelligence 2014* (Wilson *et al.*, 2014))

2. Design stochastic control policies that drive a swarm to target steady-state allocations around planar load boundaries in a way that is robust to environmental variations (*ISRR 2013* (Pavlic *et al.*, 2013a); *ASME J. of Dynamic Systems, Measurement, and Control 2014* (Pavlic *et al.*, 2015); and *Swarm Intelligence 2014* (Wilson *et al.*, 2014))

3. Compute statistical properties of our boundary coverage approach using results from computational geometry and order statistics, and formulate algorithms to generate robot configurations that maintain a connected network around a boundary (*ICRA*

*2014* (Kumar and Berman, 2014); a work under review for *IEEE Trans. on Robotics* (Kumar and Berman, 2016b))

4. Extend the analysis in Contribution 3 above to compute the properties of the random network formed by a robot team attaching to a boundary under various conditions (non-uniform coverage, distinct communication radii, non-identical spatial pdfs) and furnish relevant complexity-theoretic hardness results (a work in progress for *SIAM Journal Of Computing* (Kumar and Berman, 2016a)). Contributions 3 and 4 considerably extend the current literature on CT and BC by providing precise analytical results on the properties of this network, borrowing from extant tools in Computational Geometry, Complexity theory, and Random Geometric Graphs.

## 1.5 Outline of Dissertation

This document will comprehensively address the problem statement in Section 1.3.2 in the coming chapters. Chapter 2 reviews the literature relevant to our problem, and Chapter 3 introduces the tools and techniques used in the material in Chapter 4 through Chapter 8. The contents of these chapters are described below.

### *1.5.1 Part 1 : Ant-based CT*

Firstly, we will draw inspiration for autonomous collective transport by modeling its biological analog in the desert ant *Aphaenogaster cockerelli* in Chapter 4. This transport process is inherently *stochastic* in that ants attach and detach from the load at random time

instants. Despite having little prior knowledge of the environment, they convey the load to their nest in a fashion that is robust to changes in the size of the transport team. We will use two modeling frameworks, *Stochastic Hybrid Systems (SHS)* and *Chemical Reaction Networks (CRN)*, to rigorously model this transport phenomenon.

**Division of work**: Dr. Aurélie Buffin and Prof. Stephen Pratt (both of the School of Life Sciences, ASU) conducted the ant experiments of Section 4.2. The author devised the CRN model in Section 4.3 and derived the formulas for the moments of the state variables therein. The author also helped validate the Weighted Least Squares (WLS) algorithm of Section 4.5. Dr. Theodore Pavlic proposed the proportional velocity controller of Section 4.3 and devised and implemented the WLS algorithm.

### *1.5.2 Part 2 : Multi-boundary Coverage*

Chapter 5 details the top-down design of a stochastic control policies that drive a swarm of robots to distributed themselves along loads placed on the plane, with the object of achieving a target allocation. The robots, which start by executing correlated random walks on the plane, are driven by this controller to attach to and detach from the nearest load boundary in a stochastic manner inspired by the ants. The probability rates with which robots collide with each other and with load boundaries are called *encounter rates*, and are intractable to compute *a priori*. The stochastic control policies we develop rely instead on *ratios* between encounter rates, which are much easier to estimate, thus making tractable an otherwise nontrivial computation.

**Division of Work:** Sean Wilson and Prof. Theodore Pavlic (of the Fulton Schools of Engineering and School of Sustainability at ASU) jointly came up with the stochastic controller. Prof. Pavlic devised the CRN and the associated ODEs, and corrected this model for spatial effects arising from a non-ideal allocation of robots to a load. Sean Wilson wrote up and ran agent based simulations in NetLogo, and modified the control policies to match emulate ant behaviors in the experimentally-based model of CT addressed in Part 1. The author developed a closed-form solution to the CRN and approximated the nonlinear mean-field ODE model as a unimolecular linear model.

### *1.5.3 Parts 3 and 4: Maintaining a Wireless Network around a Boundary*

Chapter 6 approaches the boundary coverage problem from a different perspective, by considering a robot team that attaches uniformly randomly to a load boundary. We now require the robot configuration to be *connected*, meaning that every robot within a threshold distance from its two neighbors. Such a robot configuration can, for example, sense the entire load boundary, while maintaining a communication network among its members. Unlike the case of multiple load boundaries, this problem lends itself to a complete mathematical analysis using geometric methods. We devise a polynomial time algorithm to compute the probability that such a robot team connects (i.e. generates a connected communication network) upon random attachment to the boundary. We go on to describe the geometric shape of the space of connected configurations, which is then used to compute the probability density functions (pdfs) of robot positions and inter-robot distances along the boundary.

Chapter 7 substantially extends the results of Chapter 6 by considering non-uniform spatial pdfs, robots with distinct communication radii, and non-identical spatial pdfs of attachment. It also computes the properties of collision-free attachment of robots.

### *1.5.4 Part 5 : Experimental Investigation of BC with Wireless Communications*

Chapter 8 details the current state of the art in multi-robot simulation software, and the challenges involved in integrating them with wireless simulators. A comprehensive solution to this problem is left for future investigations which will address or circumvent these challenges. Our answer to this part is incomplete due to the challenges involved in developing an integrated robotic and wireless simulator.

Chapter 9 summarizes the findings from each chapter and concludes the dissertation.

# Chapter 2

# LITERATURE REVIEW

Our literature review is organized along the same lines as the problem statement.

## 2.1 Collective Transport in Robotic Swarms (Wilson *et al.*, 2014)

Current research into the problem of achieving collective transport using an autonomous robot swarm has been largely driven by the development of relatively inexpensive mobile robot platforms such as Kilobots (Rubenstein *et al.*, 2014), R-ones (McLurkin and et al., 2014), and "swarmanoid" robots (Dorigo and et al., 2013). However, some research has employed more expensive hardware with autonomous capabilities, such as quadrotors that can surveil and map their environment (Kumar and Michael, 2012). The diversity in robotic hardware has resulted in disparate approaches to collective transport. Our current work will develop robot controllers for collective transport that: (a) employ a *dynamical model* of the transport process, allowing us to analyze the transport phenomenon theoretically; (b) permit robots to *dynamically join and leave* the transport team; and (c) are *decentralized*, i.e. they do not require a centralized coordinator for their operation.

None of the extant collective transport strategies develop controllers with all three properties described above, although some strategies are characterized by one or two of the properties. For example, a dynamics-based model for transporting rigid bodies is derived and validated in (Rubenstein *et al.*, 2013). Another such physics-based model in (Stilwell and

Bay, 1993) controls the formation of a robot team to transport a palletized payload. In addition, some strategies employ a robot team that changes composition over time. For example, (O'Grady *et al.*, 2009) investigates a self-assembly application where non-functioning robots become stationary loads that must be cleared by their functional counterparts. A strategy to push a box using a transport team in which robots can reposition themselves on the load is described in (Kube and Bonabeau, 2000). Likewise, (Chen *et al.*, 2013) has a set of robots cluster behind a tall object, exploiting the object's ability to occlude a destination marker. In other approaches (Sugawara *et al.*, 2012; Becker *et al.*, 2013), the robot team merely affects the transport of the load, rather than actively propelling it to the destination; individual robots are actuated by an external control input. The work in (Berman *et al.*, 2011) models group food retrieval in ants using a hybrid system model that captures the stochastic changes to the transport team over time.

Finally, several cooperative manipulation strategies employ decentralized control, but few other than (Sun and Mills, 2002; Khatib *et al.*, 1996; Montemayor and Wen, 2005) include experimental results, and they typically ignore communication constraints. Other approaches use a centralized or leader-follower scheme that requires knowledge of load geometry (Montemayor and Wen, 2005). These centralized approaches are not scalable to large teams, particularly when the load shape is unknown.

## 2.2 Collective Transport in Ants (Kumar *et al.*, 2013)

The organization and incidence of group retrieval of food by ant teams is described in (Berman *et al.*, 2011) and (Czaczkes and Ratnieks, 2013). Group retrieval requires the

team to transport a heavy food item to their nest, the mechanisms behind which are not well understood (Franks, 1986; Czaczkes and Ratnieks, 2013). The implicit coordination among the ants in conveying the load is thought to be indirectly through the load itself, using a process called *stigmergy* (Kube and Bonabeau, 2000), although more direct interaction through pheromones may play a role. From an engineering perspective, we may describe the behavior of ants as a finite state machine (FSM), and then attempt to connect individual ants' behavior with the load dynamics. Such an FSM may help elucidate the behaviors behind successful coordination, and to compare the process of retrieval among different ant species. This model benefits both biologists and engineers: the former in understanding ant-based transport, and the latter to devise multi-robot systems for collective transport.

In (Berman *et al.*, 2011), experiments of ants collectively dragging elastic force sensors were conducted, from which qualitative observations were drawn to develop a model of the collective transport. This model was a hybrid system, consisting of both continuous and discrete parts, with probabilistic transitions between states. Our current work (Kumar *et al.*, 2013) refines (Berman *et al.*, 2011) by making quantitative observations that are incorporated into a *Stochastic Hybrid System (SHS)* model (Hespanha, 2004). We use the SHS framework to derive the statistical moments of our model's state variables, and fit these moments to our data. The SHS framework has been employed to solve problems in stochastic self-assembly of robots (Napp *et al.*, 2009), as well as stochastic task allocation (Mather and Hsieh, 2011).

## 2.3 Multi-Robot Boundary Coverage and Task Allocation (Pavlic *et al.*, 2013a; Kumar and Berman, 2014)

Previous work on decentralized multi-robot boundary coverage has focused on controlling robots to converge to uniform or arbitrary formations on a circle (Wang *et al.*, 2014). In contrast to this work, we consider cases where there is inherent and/or programmed stochasticity in the robots' motion, and our objective is to achieve robot configurations with *target statistical properties*. We assume that every robot has minimal capabilities: no global position information, and sensing or communication only within a small radius. Task allocation strategies that are suitable for such scenarios often derive robot control policies from a continuum model of the swarm population dynamics, or *macroscopic model*, in order to enable the control policies to scale with the swarm size. Various stochastic approaches to robot task allocation have focused on optimizing the task-switching rates of such macroscopic models (Correll, 2008; Liu and Winfield, 2010; Berman *et al.*, 2009; Odhner and Asada, 2010; Mather and Hsieh, 2011). Macroscopic models have also been applied to problems of robotic assembly of products, as well as robotic self-assembly (Matthey *et al.*, 2009; Evans *et al.*, 2010; Klavins *et al.*, 2006; Napp *et al.*, 2009).

One key difficulty in the development of stochastic controllers for tasks in which robots encounter each other, or features of interest, as they walk randomly in their environment is the estimation of the probability rates of these encounters, called encounter rates. These rates are nontrivial to compute analytically, but may be estimated by simulation (Hutchinson and Waser, 2007; Gurarie, 2008). Prior work in macroscopic swarm modeling (Correll and Martinoli, 2004) has employed analytic expressions for encounter rates in certain spe-

cific environments where they are easy to compute. As such, the challenge of controlling swarms in a way that precludes knowledge of encounter rates has motivated our current work (Pavlic *et al.*, 2013a, 2015; Wilson *et al.*, 2014).

## 2.4 Maintaining a Wireless Network on the Boundary

### *2.4.1 Wireless Networks as Point Processes*

The tools we use in this paper were developed by non-robotic communities. Since our interest lies is in getting our robot team to form a Wi-fi network across the boundary, we will briefly dwell on data structures for routing in Mobile Ad Hoc Networks *MANETs* (Jamalipour and Ma, 2011). However, our main thrust lies in analyzing the properties of the Geometric Graph (GG) formed by the robot team's network. Our probabilistic analysis borrows heavily from the formalism of *Poisson Point Processes (PPP)* (Penrose, 2003) a class of spatial stochastic processes in which each robot takes positions independently of others. The network induced by a PPP is a *Random Geometric Graph* (RGG) (Penrose, 2003). When attachments are required to be *collision-free* (CF), i.e. have no colliding pairs of robots, attachments are characterized by a Matern hard core process (HCP). These spatial processes and their resulting RGGs have been heavily used in the Wireless communication literature (Haenggi, 2012).

### 2.4.2 Computational Geometry

Many of our probabilistic results involving the communication graph $\mathscr{G}$ involve computing the volume of a convex polytope, or an integral over it. Our hardness result in Section 7.7 has its roots in the work of Dyer, Frieze and other computation theorists in determining and approximating the polytope volume (Dyer and Frieze, 1991, 1988).

Our results for the SCS involving Renyi parking stems from the work of Renyi (Rényi, 1958), Dvoretzky et.al. (Dvoretzky and Robbins, 1964), and other probabilists. The Renyi Parking Problem (RPP) defined in Section 7.5 has been extensively studied in the physics literature under the name *Random Sequential Adsorption (RSA)*, the process by which molecules get adsorbed onto a substrate surface (Talbot *et al.*, 2000; Sikiric and Itoh, 2011). The delay differential equation (DDE) governing the mean number of parked cars is extensively analyzed herein ; moreover, (Sikiric and Itoh, 2011) computes the asymptotic properties of an interval tree storing the occupied subintervals of the parking lot. To our knowledge, however, there has been no analysis of the spatial pdfs generated by the parking problem. We furnish an algorithm to compute the joint spatial pdf as the ratio of volumes of two polytopes. This ratio has no compact closed form that we are aware of, yet it can be computed in pseudo-$\mathsf{P}$ time.

## 2.5 Multi-robot and Wi-fi Simulation

We have investigated the advantages and drawbacks of two robot simulators: ROS (Open Source Robotics Foundation, 2015) and ARGoS (Pinciroli *et al.*, 2012).

### *2.5.1 ROS*

*ROS(Robot Operating System)*, developed by the Open Source Robotics Foundation (OSRF), is a robotic simulator that is generally meant to be used for single robots. A large number of robotic algorithms have been implemented and tested on ROS, for example the AMCL Package (Gerkey, 2015) implements the Adaptive Monte Carlo Localization Method (Thrun *et al.*, 2005); the hector SLAM package (Kohlbrecher and Meyer, 2015) implements the SLAM algorithm of (Kohlbrecher *et al.*, 2011), and the Laser Scan Matcher Package (Ivan Dryanovski, William Morris, Andrea Censi, 2015) implements the Iterative Closest Points algorithm from (Censi, 2006). ARGoS (Pinciroli *et al.*, 2012) is a recently developed multirobot simulator.

The core of ROS is written primarily in C++, and is exposed by a set of C++and Python APIs. Multi-Robot simulations are supported by the *Rocon* (ROS in concert) (Stonier *et al.*, 2013) package, in which a multi-robot team is represented by multiple ROS *masters*, where every master represents the software process of a distinct robot. Multiple masters running on the same network can intercommunicate using the ROS publish/subscribe architecture. Each robot can be visualized on a separate instance of the Gazebo graphical simulator (Gazebo, 2016). ROS has modules for kinematics, dynamics, path planning, localization and computer vision. However, ROS does not handle wireless communication by itself.

### *2.5.2 ARGoS*

Unlike ROS, ARGoS was designed to be a multi-robot system from its inception. ARGoS is less mature than ROS, and had its first release only in 2012. ARGoS consists of controlling, actuating, sensing, and dynamical modules (Pinciroli *et al.*, 2012), but lacks capabilities such as motion planning, localization and Kalman filtering that are characteristic of ROS' navigation system. Its strength lies in its support of swarms of up to 100,000 robots (Pinciroli *et al.*, 2012); indeed, ARGoS ' basic examples emphasize the collective behavior of swarms. An earlier version of ARGoS has been patched with NS3 through RoboNetSim (Kudelski *et al.*, 2013) to yield a multi-robot system that communicates through a Wi-fi network. At the time of this writing, a similar patch needs to be developed for the latest version of ARGoS to implement the design in Section 8.1.

### *2.5.3 NS3*

NS-3 is a discrete event simulator , whose core is written in C++. Just like ROS, NS-3 provides both C++and Python APIs. NS-3 allows us to create and define the Mobile Ad Hoc Network of a multi-robot team, and route messages between robots using protocols such as Open Link State Routing (OLSR) and Ad Hoc On Demand Vector Routing (AODV). Also, NS-3 models wireless effects such as propagation loss and multipath fading with excellent fidelity (Renard *et al.*, 2012).

### 2.5.4 Pylayers

Pylayers (Amiot *et al.*, 2013) is an indoor radio channel simulator for mobile heterogeneous networks. It was developed mainly to model the propagation of radio signals between humans wearing Wi-fi devices as they move within a building. Pylayers captures such a movement scenario in four *layers* of simulation: (1) a Layout layer, which details the simulation scene, such as the rooms of the building, visibility graph, and the topology graph that determines the propagation paths of radio signals, (2) a Mechanical layer, which describes the movement of human agents, (3) a Network layer, which details radio interactions among the human agents, and (4) a Localization layer, which estimates the position of each agent based on the signals they transmit and receive.

### 2.5.5 GENI

GENI (Global Environment for Network Innovation) is a virtual lab for computer networks and distributed systems (Berman *et al.*, 2014) supported by the NSF. Our interest in GENI stems from the fact that the project allots wireless sensor network testbeds to researchers, on which routing experiments can be run (Chen *et al.*, 2012). GENI hardware has two advantages: it is deeply programmable, thereby precluding the need to manually install software (such as DHCP) on each node, and does not rely on software simulation of wireless channels.

# Chapter 3

# BACKGROUND

This chapter furnishes the background for the theory developed later in the dissertation.

## 3.1 Chemical Reaction Networks (CRNs)

Our interest in Chemical Reaction Networks (CRNs) stems from their use in modeling robots as species in a chemical reaction. The behavior of a multi-robot team over time can then be concisely written as a chemical reaction, and the laws of mass action kinetics may be employed to determine the proportion of robots in each behavioral state as a function of time. We will use CRNs to model the behavior of ants engaged in CT in Chapter 4. The material in this section is paraphrased from (Gillespie, 1992, 2007).

CRN theory is a branch of applied mathematics that models chemical reactions as nonlinear ordinary differential equations (ODEs). A chemical reaction consists of a set of reactants that give rise to products, and a CRN is a set of multiple reactions. Each reaction has a *rate constant* that indicates the probability that the reactant will change into the product in unit time. Let $\mathbf{x}$ be a vector of concentration of complexes as a function of time. The evolution of $\dot{\mathbf{x}}$ can be written as a nonlinear ODE of the form

$$\dot{\mathbf{x}} = \Gamma \mathbf{V}(\mathbf{x}) \tag{3.1}$$

called the rate equation. where $\Gamma$ is the *stoichiometric matrix* and $\mathbf{V}(\mathbf{x})$ is a function of $\mathbf{x}$ derived from mass-action kinetics. A CRN is said to reach steady state when $\dot{\mathbf{x}} = 0$. A CRN consisting of a pair of reversible reactions reaches steady state when the rates of the forward and backward reaction are equal. The reaction

$$a_1\mathrm{R}_1 + \ldots + a_m\mathrm{R}_m \xrightarrow{k} b_1\mathrm{P}_1 + \ldots + b_n\mathrm{P}_n. \tag{3.2}$$

has the rate equation

$$\dot{\mathbf{x}} = k\, R_1^{a_1} R_2^{a_2} \ldots R_m^{a_m} \text{ where } \mathbf{x} = \begin{bmatrix} R_1 & \ldots & R_m \end{bmatrix}^T. \tag{3.3}$$

**Example 3.1.1.** *The pair of reversible reactions*

$$\mathrm{A} \xrightarrow{k_1} \mathrm{B}, \;\; \mathrm{B} \xrightarrow{k_2} \mathrm{A} \tag{3.4}$$

*is a CRN. The species involved are* A *and* B*, and the complexes are the same as the species. The vector of species concentrations is given by* $\mathbf{x} = [A\ B]^T$*, where A and B denote species concentrations. The evolution of this pair of reactions is governed by*

$$\dot{\mathbf{x}} = \begin{bmatrix} -k_1 & k_2 \\ k_1 & -k_2 \end{bmatrix} \mathbf{x}, \tag{3.5}$$

*The CRN will reach equilibrium when the rates of the two reactions are equal, i.e. when* $k_1 A = k_2 B$.

## 3.2 Stochastic Hybrid Systems

The concept of Finite State Machines (FSMs) is familiar from computation theory. FSMs have been extended in two significant ways by control theorists. Firstly, when states are

allowed to be continuous as well as discrete, the resulting automaton becomes a *Hybrid Automaton*. Secondly, when either the discrete or the continuous states have a stochastic element in them, the resulting automaton is called a Stochastic Hybrid System (SHS) (Hespanha and Singh, 2005). Formally an SHS is defined by a Stochastic Differential Equation (SDE) of the form

$$\dot{\mathbf{x}} = f(\mathbf{q},\mathbf{x},t) + g(\mathbf{q},\mathbf{x},t)\dot{\mathbf{n}}, \tag{3.6}$$

a family of reset maps

$$\phi_l : (\mathbf{q},\mathbf{x}) \rightarrow (\mathbf{q}_{\text{new}},\mathbf{x}_{\text{new}}), \text{ where } l \in 1,\ldots,m. \tag{3.7}$$

and a family of transition intensities

$$\lambda_l(\mathbf{q},\mathbf{x},t), \text{ where } l \in 1,\ldots,m. \tag{3.8}$$

Here, $\mathbf{x}$ is a stochastic process in $\mathbb{R}^n$ with piecewise continuous paths denoting the continuous state if the SHS, $\mathbf{q}$ denotes the discrete state taking values within a discrete set $Q$ (corresponding to the states of an FSM), and $\mathbf{n}$ is a vector of $k$ independent Brownian motion processes. The time $t$ is a nonnegative real number.

The time evolution of an SHS is determined by its extended generator operator $L$. Suppose $\psi(\mathbf{q},\mathbf{x},t)$ is a function of the SHS state whose expectation we wish to compute. If $\psi$ is twice continuously differentiable with respect to $\mathbf{x}$ and once continuously differentiable with respect to $t$, we have the identity

$$\frac{d}{dt}\mathrm{E}(\psi(\mathbf{q},\mathbf{x},t) = \mathrm{E}(L\psi(\mathbf{q},\mathbf{x},t)), \text{ for all } (\mathbf{q} \in \mathbb{Q}, \mathbf{x} \in \mathbb{R}^n, t \in \mathbb{R}_+). \tag{3.9}$$

Here $L$ is called the *extended generator operator*. The definition of $L$ is left to (Hespanha and Singh, 2005). An SHS is said to be a *polynomial SHS (pSHS)* iff its $L$ is a finite polynomial whenever $\psi$ is a finite polynomial in its arguments. Every CRN has an SHS corresponding to it, as we illustrate with Example 3.1.1.

**Example 3.2.1.** *Let* $\mathbf{x} = [A\ B]^T$ *denote the vector of concentrations as in Example 3.1.1. The associated SHS has the continuous dynamics* $\dot{\mathbf{x}} = 0$*, and two reset maps and transition intensities, one for each reaction:*

$$\mathrm{A} \xrightarrow{k_1} \mathrm{B} : \phi_1(\mathbf{x}) = \begin{bmatrix} A-1 & B+1 \end{bmatrix}^T, \lambda_1 = k_1 A \tag{3.10}$$

$$\mathrm{B} \xrightarrow{k_2} \mathrm{A} : \phi_2(\mathbf{x}) = \begin{bmatrix} A+1 & B-1 \end{bmatrix}^T, \lambda_2 = k_2 B \tag{3.11}$$

## 3.3 Computational Geometry

A geometric shape $\mathscr{P}$ embedded in $\mathbb{R}^n$ is called a *polytope* if it is bounded on all sides by hyperplanes; it is *convex* if it can be expressed as the intersection of half-spaces (Grünbaum, 2002, Chap.1). A convex polytope specified thusly is said to be in $H$ (hyperplane) form. It can be specified as the convex hull of its vertices instead in which case it is said to be in the $V$ (vertex) form.

A polytope $\mathscr{S}$ with $n+1$ vertices embedded in $\mathbb{R}^n$ is called a *simplex*. We will commonly specify a simplex using its *vertex matrix*, $\mathbb{V}(\mathscr{S})$, each of whose columns gives the coordinates of one of the vertices. The volume of the simplex $\mathrm{Vol}(\mathscr{S})$ can be determined by the *Cayley-Menger* determinant (Muncherino *et al.*, 2013, p.34) of $\mathbb{V}(\mathscr{S})$. Topologically, the vertices of $\mathscr{S}$ form a clique, with every pair of vertices being connected by an edge. The vertices in $\mathbb{V}(\mathscr{S})$ define the convex hull of $\mathscr{S}$, since every point in the interior of $\mathscr{S}$ can

be represented as a convex combination of these vertices. The simplex

$$\Delta_n := \{(x_1, x_2, \ldots, x_n) \in \mathbb{R}^n : x_i \geq 0 \text{ and } \sum x_i \leq 1\} \tag{3.12}$$

is called the *canonical simplex* in $\mathbb{R}^n$.

We may partition any convex polytope into simplices by a process called *simplicial subdivision*. We can then sum the volumes of the subdividing simplices to determine the volume of the polytope. One possible simplicial subdivision is created by adjoining one interior point of $\mathscr{P}$ to each facet.

## 3.4 Order Statistics

Suppose $X_{i:1\leq i\leq n}$ are $n$ iid random variables each having pdf $h$ and cdf $H$, each supported on a interval $\mathscr{B}$ of $\mathbb{R}$. Let their realizations be $\mathbf{x} = \begin{bmatrix} \mathtt{x}_1 & \ldots & \mathtt{x}_n \end{bmatrix}^T$, where $x_i \in \mathscr{B}$. The quantities in $\mathtt{x} = \begin{bmatrix} \mathtt{x}_i \end{bmatrix}^T$ generated by sorting $\mathbf{x}$ in nondecreasing order are called the *order statistics* of $\mathbf{x}$. The unordered variables $X_i$ are called *parent variables*, and $h$ and $H$ are respectively called the *parent pdf and cdf*. The ordered position $\mathtt{X}_i$ is called the $i$-th *order statistic* of the parent variable $X_i$ and is the $i$-th smallest of the $n$ positions in $\mathbf{X}$. In general, if the $n$ components of $\mathbf{X}$ are i.i.d variates, each having pdf $f(t)$ and cdf $F(t)$, then we have the following expressions for the joint and marginal pdfs (David and Nagaraja, 2003, p. 9):

$$\begin{aligned} f_{\mathtt{X}}(\mathtt{x}) &= n! \prod_{i=1}^{n} f(t) \mathbf{1}_{\mathscr{P}}, \\ f_{\mathtt{X}_i}(\mathtt{x}_i) &= \sum_{j=i}^{n} \binom{n}{j} F(t)^j (1-F(t))^{n-j}. \end{aligned} \tag{3.13}$$

where $\mathscr{P}$ is that subregion of $\mathbb{R}^n$ consisting solely of points with nondecreasing coordinates

$$\mathscr{P} := \{\mathtt{x} \in \mathbb{R}^n : (\mathtt{x}_1, \ldots, \mathtt{x}_n) : \mathtt{x}_1 \leq \mathtt{x}_2 \ldots \leq \mathtt{x}_{n-1} \leq \mathtt{x}_n\} \tag{3.14}$$

(David and Nagaraja, 2003, Chap.2) also provides expressions for the joint pdf and cdf of a subvector of $\mathbf{X}$.

The special case in which $\mathscr{B}$ is a finite subinterval of $\mathbb{R}$ is of special interest, for we may think of it as a (finite) boundary to which point robots attach, at positions $\mathbf{x}$. The resulting $\mathscr{P}$ becomes a *simplex* in the positive orthant of $\mathbb{R}^n$, and has the form

$$\mathscr{P} := \{\mathtt{x} \in \mathbb{R}^n : (\mathtt{x}_1, \ldots, \mathtt{x}_n) : 0 \leq \mathtt{x}_1 \leq \mathtt{x}_2 \ldots \leq \mathtt{x}_{n-1} \leq \mathtt{x}_n \leq s\}, \tag{3.15}$$

where $s$ is the boundary length.The marginal pdfs of order statistics, as well as the joint pdfs of any of their subsets, can be computed in principle by a multivariable integral over $\mathscr{P}$. To make formulas easier to state, we will define the artificial positions $\mathtt{x}_0$ and $\mathtt{x}_{n+1}$ to be the left and the right endpoints of $\mathscr{B}$.

### *3.4.1 Probability Integral Transform*

It is generally easier to compute with uniform parents rather than arbitrary nonuniform ones. Suppose $X_i$ are nonuniform iid variates supported on $\mathscr{B}$. Define the random variables $Y_i$ by

$$f_{Y_i}(x) := H_{X_i}(x)\mathbf{I}_{Bo}. \tag{3.16}$$

The pdf of $Y_i$ takes on the value of the cdf of $X_i$ at every point on $\mathscr{B}$. The variable $Y_i$ is called the *probability integral transform* of $X_i$.

### 3.4.2 Order Statistics for Non-IID Parents

While we will encounter inid parents only briefly in Chapter 7, it is worthwhile to point out that computing their order statistics is computationally expensive, as given by

**Theorem 3.4.1.** ***Bapat-Beg Theorem*** *(David and Nagaraja, 2003, Ch.5), (Glueck* et al.*, 2008): Given the inid pdfs* $\mathbf{h}_{1:n}$ *with cdfs* $\mathbf{H}_{1:n}$ *supported on* $\mathbb{R}$*. Then the order statistics of the vector* $\mathbf{x}_{1:n}$*, where* $\mathtt{X}_i \sim h_i(x)$ *have the joint cdf given by*

$$F_{\mathtt{X}}(\mathtt{x}) = \text{per}(\mathbb{L}(\mathtt{x}))\mathbf{I}_{\mathscr{D}} \tag{3.17}$$

*where* $\mathbb{L}_{n\times n}$ *has the entries*

$$\mathbb{L}_{i,j} = H_j(\mathtt{x}_i) - H_j(\mathtt{x}_{i-1}) \tag{3.18}$$

The first row $\mathbb{L}_{:,j}$ consists of entries of the form $H_j(\mathtt{x}_1) - H_j(\mathtt{x}_0) = H_j(\mathtt{x}_1)$. We will call $\mathbb{L}$ the *cdf matrix* of the functions $\mathbf{H}$.

## 3.5 Counting Complexity

We will now define the counting complexity classes #P (Sharp-P), which pertains to *counting problems* that ask "How many elements of a given space satisfy a desired property?" (Jerrum *et al.*, 2004). For example, the counting version of the satisfiability problem #SAT supplies a boolean formula $\Phi$ as input and asks for the number of satisfying assignments to $\Phi$ as output.

**Definition 3.5.1.** *A counting Turing machine is a standard nondeterministic TM with an auxiliary output device that prints the number of accepting paths induced by the input on*

*a special tape in binary format. Its time complexity is defined to be $f(n)$ iff the longest accepting computation induced by inputs of length n takes $f(n)$ steps.* #P *is the class of functions that can be computed by counting TMs of polynomial time complexity (Valiant, 1979).*

We define the classes #PH (Sharp-P-Hard) and #PC (Sharp-P-Complete) analogously to NPH and NPC are defined in terms of NP. The first problem to be proved #PH was that of computing permanents:

### *3.5.1 Computing Permanents is* #PH

Let $\mathbb{M}_{m\times m}$ be a square matrix. Let per$(.)$ denote the permanent of the square matrix in its argument. Define two complexity-theoretic problems, PERMINT and PERMBOOL as follows: PERMINT (resp. PERMBOOL) takes in a integer (resp. Boolean) $m \times m$ matrix as input, and asks for its permanent as output. **Valiant's theorem** states that PERMINT is #PH, and PERMBOOL is #PC. When $\mathbb{M}$ consists of nonnegative entries (i.e. $M_{i,j} \geq 0$), per$(\mathbb{M})$ is P-approximable to arbitrary accuracy.(Valiant, 1979; Jerrum *et al.*, 2004).

The counting versions of all known NPC problems are #PC, for example those of 0-1 Knapsack, 3-Cover, and Bipartite Matching. Our problems involving computing probabilities that are the ratio of volumes of polytopes will require the following results.

## 3.6 Generalized Simplex Hypercube Intersection

In Chapter 7, we will demonstrate that our problems involving computing the connectivity of a certain graph are #PH by reduction from the VHSP problem taken from (Dyer and Frieze, 1988, 1991). Define the problem VHSP as one that computes the volume of intersection between a half-space and the unit hypercube,with

Input: Parameters $\mathbf{a}_{1:n}, b$ of the halfspace $\mathscr{T} := \{\mathbf{s} \in \mathbb{R}^n : \mathbf{a}^T \mathbf{s}_{1:n} \leq b\}$, with the entries of $\mathbf{a}$ and $b$ being positive rationals

Output: Volume of intersection of $\mathscr{T}$ with the unit hypercube $\mathscr{C} := [0,1]^n$

**Lemma 3.6.1.** *The solution to* VHSP *given by*

$$\mathrm{Vol}(\mathscr{T} \cap \mathscr{C}) = n! \prod_{i=1}^{n} \sum_{\mathbf{v} \in \{0,1\}^n} (-1)^{\mathbf{1}^T \mathbf{v}} \max((b - \mathbf{a}^T \mathbf{s})^n, 0). \tag{3.19}$$

*is* #PH. *Equivalently, it is* #PH *to find the probability that a random point in* $\mathscr{C}$ *satisfies a single linear inequality.*

### 3.6.1 Recasting VHSP as a Simplex Hypercuboid Intersection

It will be useful to redefine VHSP as an intersection between a half-space with unit coefficients and a generic hypercuboid. Introduce the primed variables $s'_i := a_i s_i$, under which transformation the images of $\mathscr{T}$ and $\mathscr{C}$ become

$$\mathscr{T}' := \{\mathbf{s}' \in \mathbb{R}^n : \mathbf{1}^T \mathbf{s}' \leq b\} \text{ and} \tag{3.20}$$

$$\mathscr{C}' := \prod [0, a_i]. \tag{3.21}$$

Let $V := \text{Vol}(\mathscr{T} \cap \mathscr{C})$ and $V' := \text{Vol}(\mathscr{T}' \cap \mathscr{C}')$. Using the change of variables formula for multivariable integrals (Zill *et al.*, 2011), we have that $V = \det(J)V'$, where $J_{n\times n}$ is the Jacobian matrix of the transformation, given by

$$J_{i,j:1\leq i,j\leq n} := \frac{\partial s_i}{\partial s'_j} \tag{3.22}$$

Evidently, the only nonzero elements of $J$ lie on its diagonal, with $J_{i,i} = \frac{\partial s_i}{\partial s'_i} = \frac{1}{a_i}$, so $\det(J) = \prod_{i=1}^{n} \frac{1}{a_i}$, yielding

$$V = V'(\prod_{i=1}^{n} \frac{1}{a_i}). \tag{3.23}$$

This formula recovers the volume $V$ from the transformed volume $V'$ in polynomial time.

**Example 3.6.1.** *Consider the intersection between the halfspace* $\mathscr{T} := \{(s_1, s_2) : 2s_1 + 3s_2 \leq 2\}$ *and the unit hypercube* $\mathscr{C} : \{(s_1, s_2) \in [0,1]^2\}$. *We have* $a_1 = 2, a_2 = 3$ *and* $b = 2$.*The region of overlap* $\mathscr{T} \cap \mathscr{C}$ *is a triangle with vertices* $(0,0), (1,0)$ *and* $(0, \frac{2}{3})$ *and area* $V := \frac{1}{3}$.

*Rescale the axes using* $s'_1 := a_1 s_1 = 2s_1$ *and* $s'_2 := a_2 s_2 = 3s_2$. *In the rescaled coordinate system, the halfspace and the hypercuboid (rectangle) become*

$$\mathscr{T}' := \{(s'_1, s'_2) : s'_1 + s'_2 \leq 2\} \tag{3.24}$$

$$\mathscr{C}' := \{(s'_1, s'_2) : 0 \leq s'_1 \leq 2, \text{ and } 0 \leq s'_2 \leq 3\}. \tag{3.25}$$

*The region of overlap is a triangle with vertices at* $(0,0), (2,0)$ *and* $(0,2)$ *and area* $V' := 2$. *The Jacobian has determinant* $\det(J) = \frac{1}{a_1 a_2} = \frac{1}{6}$, *so that* $V = V'/6$.

**Table 3.1:** Complexity Classes and Problems

| Class / Problem Name | Expansion |
|---|---|
| NPH, NPC | NP-Hard, NP-complete |
| #PH, #PC | Sharp-P-Hard, Sharp-P-Complete |
| PERMBOOL/PERMINT | Permanent of Boolean / Integer Matrix |
| #SAT, #KNAP | Counting versions of SAT and and 0-1 Knapsack |
| VHSP | Volume of simplex-hypercube intersection |

The Markov Chain Monte Carlo (MCMC) algorithm is a *Fully Polynomial Randomized Approximation Scheme (FPRAS)* for computing the volume of a polytope as well as sampling a point uniformly from it (Jerrum *et al.*, 2004).

Table 3.1 summarizes all the complexity classes and problems introduced thus far.

## 3.7 Poisson Point Processes (PPP) and Random Geometric Graphs (RGG)

We will now consider a team of point robots setting up a wireless network on a boundary. Suppose the boundary is the interval $\mathscr{B} := [0, s]$ and point robots take up positions at random on it, just as given in Section 3.4. For the robots to form a connected network across the boundary, we need every robot to have a neighbor situated at a distance $d$ or less away from it, where the connectivity threshold $d$ represents the communication radius of a robot. With this setting in mind, we borrow the following concepts from the geometry and probability.

### 3.7.1 Random Geometric Graphs

Suppose $n$ points $\mathbf{x}_{1:n}$ are chosen randomly in $\mathbb{R}^d$, with each point being generated by the iid random variables $\mathbf{X}_{1:n}$. Suppose a connectivity threshold $d$ is prescribed. Define a graph $\mathscr{G}$ with vertices at $\mathbf{x}$, and two vertices connected by an edge iff their Euclidean distance lies within $d$. The resulting graph is called a *Random Geometric Graph (RGG)* (Penrose, 2003). The properties of the RGG, such as the number of clusters, the probability of its connectivity, and the longest distance between neighbors are hard to compute exactly, though their asymptotics have been determined (Penrose, 2003).

We borrow the following definitions from (Haenggi, 2012).Define a *Point Process (PP)* in $\mathbb{R}^d$ as a countable random collection of points in it. The associated sigma algebra is the collection of Borel sets in $\mathbb{R}^d$, and the measure is the Lebesgue measure. We may also describe a PP by the number of points it places in every subset $B$ of $\mathbb{R}^d$, and call this quantity $N(B)$. We define the intensity measure of the PP to be

$$\Lambda(B) = \mathbb{E}(N(B)) \; \forall B \subseteq \mathbb{R}^d. \tag{3.26}$$

A Poisson Point Process of intensity $\Lambda$ in $\mathbb{R}^d$ is a PP with the properties that:

1. For every compact subset $B$ of $\mathbb{R}^d$, the number of points $N(B)$ placed in $B$ by the PPP is a Poisson random variable with mean $\Lambda(B)$.

2. If $B_{i:i=1,\ldots,n}$ are disjoint compact subsets of $\mathbb{R}^d$, then the random variables $N(B_i)$ are independent.

If $\Lambda$ has a density $\lambda$, then we have

$$\mathbb{P}(N(B) = k) \sim \text{Poi}(\int_B \lambda(x)dx) \tag{3.27}$$

### 3.7.2 PPPs in $\mathscr{B}$

We are most interested in PPPs in a single dimension, for our problem requires wireless communication on a finite 1-dimensional boundary $\mathscr{B}$. A homogeneous PPP on $\mathscr{B}$ places points uniformly on $\mathscr{B}$, so that the expected number of points in the subinterval $[a,b]$ of $\mathscr{B}$ is a Poisson variate with parameter $b-a$. This PPP corresponds to the case of the uniform parent of Section 3.4. Thus, the count $N(B)$ where $B = [a,b]$ depends only on the length of $B$, and not on the location of $B$. The homogeneous PPP places equal numbers of points on average on subintervals of equal length.

On the other hand, every non uniform parent on $\mathscr{B}$ corresponds to a non-homogeneous PPP, with the counts being equal to

$$\mathbb{P}(N([a,b]) = k) \sim \text{Poi}(\int_a^b \lambda(x)dx) \tag{3.28}$$

Since the intensity $\lambda$ is nonconstant, the values of $N(B)$ will vary with the location of the left end-point $a$, as well as with the length $b-a$ of the subinterval.

### 3.7.3 Matern Hard Core Processes in $\mathscr{B}$

We will need to consider the attachments of finite robots to the boundary in Section 7.5 in the context of the Renyi Parking Problem. These robots each have identical diameter $R$.

Typically, the attachments need to be *conflict-free*, i.e. preclude conflicts between adjacent robots. A random conflict-free configuration of robots is the realization of a *Type III Matern Hard-core process (HCP)*. We realize such a Type III HCP in $\mathscr{B}$ by generating $n$ random points $\mathbf{x}_{1:n}$ on $\mathscr{B}$, using a PPP , and marking each with a unique id chosen from $1 \ldots n$. Subsequently, we visit the points in increasing order of their id and check for conflicts. Every point $x_{j:j>i}$ that collides with $x_i$ is removed in the course of visiting $x_i$. The resulting configuration has $m \leq n$ robots, and is the realization of a Type III HCP.

Chapter 4

# COLLECTIVE TRANSPORT IN THE DESERT ANT *A. COCKERELLI*

**Source:** (Kumar *et al.*, 2013)

## 4.1 Abstract

Collective food transport in ant colonies is a striking, albeit poorly understood, example of coordinated group behavior in nature that can serve as a template for robust, decentralized multi-robot cooperative manipulation strategies. We investigate this behavior in *Aphaenogaster cockerelli* ants in order to derive a model of the ants' roles and behavioral transitions and the resulting dynamics of a transported load. In experimental trials, *A. cockerelli* are induced to transport a rigid artificial load to their nest. From video recordings of the trials, we obtain time series data on the load position and the population counts of ants in three roles. From our observations, we develop a stochastic hybrid system model that describes the time evolution of these variables and that can be used to derive the dynamics of their statistical moments. In our model, ants switch stochastically between roles at constant, unknown probability rates, and ants in one role pull on the load with a force that acts as a proportional controller on the load velocity with unknown gain and set point. We compute these unknown parameters by using standard numerical optimization techniques to fit the time evolution of the means of the load position and population counts to the averaged experimental time series. The close fit of our model to the averaged data and to data

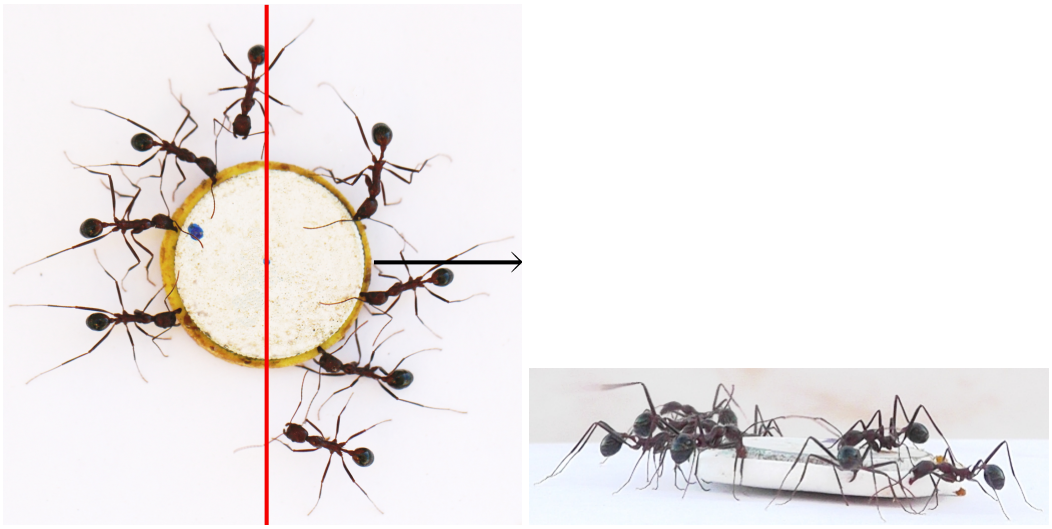
for individual trials demonstrates the accuracy of our proposed model in predicting the ant behavior.

## 4.2 Experimental Trials

We filmed colonies of *A. cockerelli* collectively retrieving a standardized artificial load. A total of 17 colonies were located in South Mountain Park in Phoenix, AZ. Experiments were carried out during the activity period of the colony in the early morning (0600–0830 hours) and in the late afternoon (1700–1900 hours) in May 2012.

A new colony was located each day. A Plexiglas® sheet with dimensions 61 cm × 46 cm × 0.5 cm was positioned such that one edge, Edge A, was 50 cm south of the main nest entrance, and the opposite edge, Edge B, was 111 cm south of the nest. The sheet was covered with white paper and leveled to avoid inclination in any direction. An artificial load was constructed by gluing a dime with mass $m_L = 2.30$ g and radius 0.90 cm to an ethylene vinyl acetate (EVA) foam disk (0.2 mm thickness, 1.0 cm radius), which was rubbed with fig paste to attract ants. We filmed the transport with a Canon G12 camera positioned above the sheet. The camera's field of view was 1280 pixels × 720 pixels, centered on the sheet. Foragers were recruited to a whole fig placed at Edge B. Once 10 workers were feeding on the fig, we replaced the fruit with the artificial load. Ants were able to manipulate the load by gripping the excess 0.1 cm of foam around the perimeter of the dime. Ants carrying the load were filmed until they reached Edge A.

From the video recording of each experimental trial, we selected a segment of duration 145 s during which the ants were smoothly transporting the load; i.e., the load moved a



(a) Top view (b) Side view

**Fig. 4.1:** *Aphaenogaster cockerelli* ants transporting an artificial load: (a) top view, with the arrow indicating the direction of the load motion and the red line dividing the load into *front* (right) and *back* (left) halves; (b) side view. The views are from different trials.

nonzero distance during each consecutive 5-second interval of the segment. From this segment, we extracted the positions of the ants around the load and the position of the center of the load using ImageJ (Schneider *et al.*, 2012) and the Mtrack plug-in (Meijering *et al.*, 2012). This information was obtained from single frames at 5-second intervals.

This data revealed a number of interesting details about the transport process. In each segment, the load was transported approximately along a straight line, and at approximately the same speed. We next divided the load into two halves perpendicular to the direction of motion, as shown in the top view Figure 4.1((a)), calling them the *front* and *back* halves. As a result, each ant was either detached from the load at a given time instant, or was attached to one of the halves. We call the corresponding behavioural states *Detached, Front*, and *Back* respectively, and define

$$S := \{F(\text{front}), B(\text{back}), D(\text{detached})\} \tag{4.1}$$

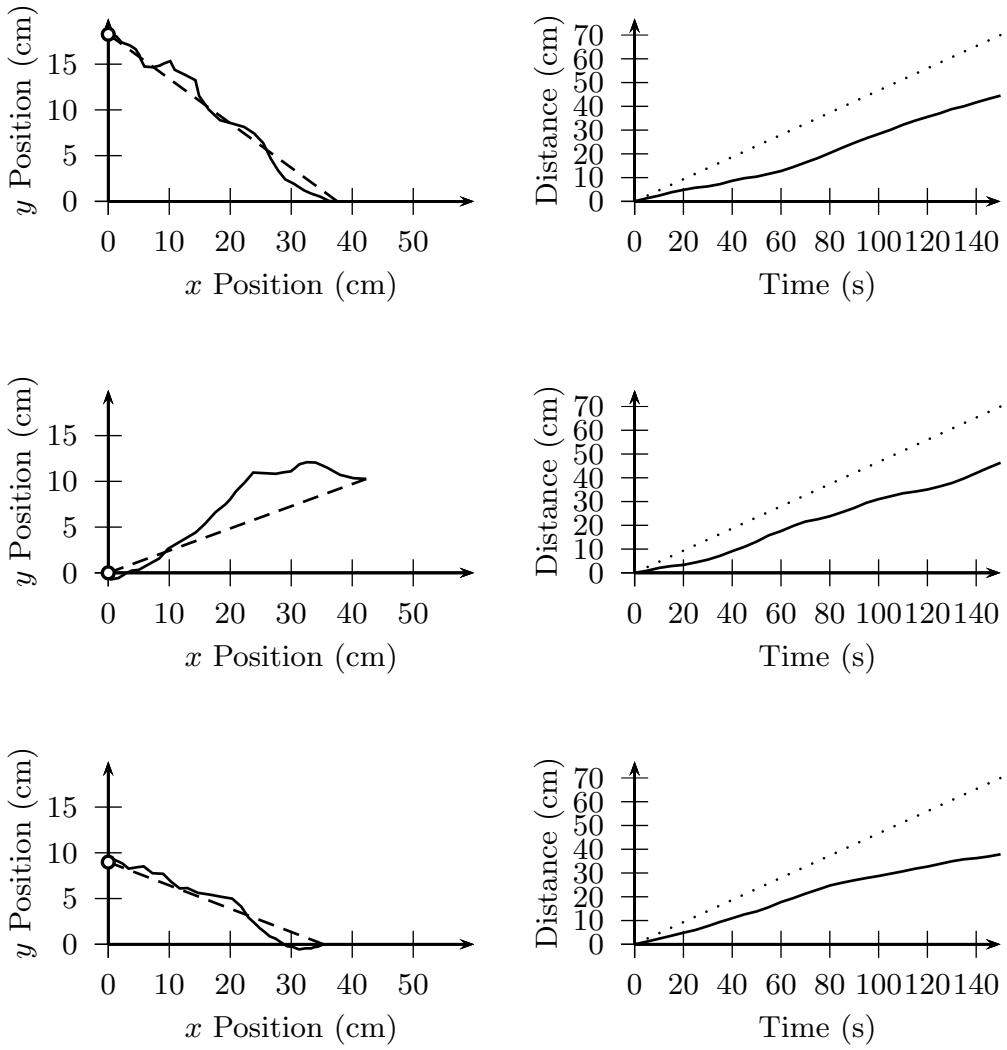



**Fig. 4.2:** The *x*–*y* trajectory of the artificial load center (left column) and the distance traveled by the load over time (right column) for three experimental trials. The dashed lines in the left column show the ideal straight-line path. The dotted reference lines in the right column all have the same slope (Kumar *et al.*, 2013).

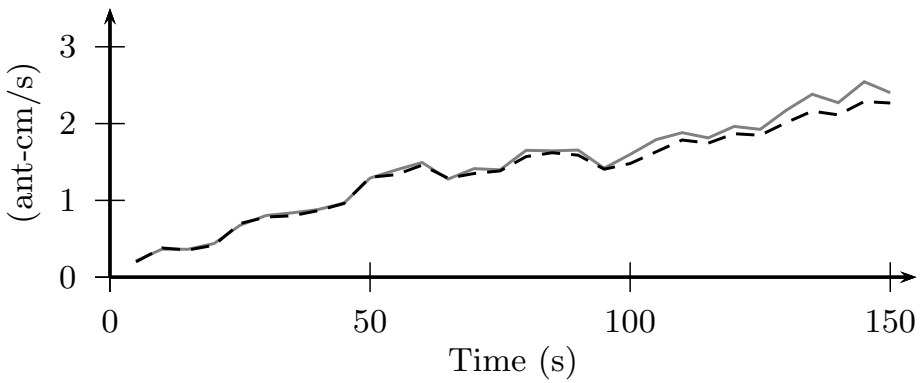



**Fig. 4.3:** The solid line shows the sample mean of the product $N_F v_L$ from the data; the dashed line shows the product of the sample mean of $N_F$ with the sample mean of $v_L$ from the data.

to be the set of states. Also, from the side view in Figure 4.1((b)), we noticed that every ant lifted the load, with the front ants pulling it towards their nest in addition. We estimated that the mean lifting force exerted by each ant was $F_L = 2.653$ mN in a separate experiment. Next, we covered an inclined plane with the same paper as was used in the ant experiments, and measured the angle at which the load started to slide down it as $\theta_s = 30^\circ$. This provided an estimate of the coefficient of kinetic friction of the sheet as $\mu = \tan\theta_s = 0.58$.

## 4.3 Stochastic Hybrid System Model

**Modeling Behavior** We first model the behavioral component as a three-state probabilistic finite state machine with the set of states in Equation (4.1). Every transition between states is characterized by a *transition rate*, which is the probability that the transition will occur within unit time. This FSM has a transition from every state to every other state, having the form

$$S_i \xrightarrow{r_{ij}} S_j \text{ where } S_i, S_j \in S, \text{ and } S_i \neq S_j, \tag{4.2}$$

where $r_{ij}$ denotes the transition rate of switching from $S_i$ to $S_j$. For example, $r_{BD}$ denotes the rate of switching from the *Back* state to the *Detached* state. The six transition rates $r_{ij}$ are undetermined. Since Equation (4.2) describes the state transitions of *individual ants*, we call it the *microscopic* model of the SHS.

Consider a single ant in state $S_i$ to be a molecule of the species $S_i$. Observe that each transition in Equation (4.2) is a *unimolecular* chemical reaction, i.e. it has a single molecule of the reactant $S_i$. Define the *transition intensity* of the reaction Equation (4.2) by $\lambda_{ij} := r_{ij}N_i$ (Hespanha and Singh, 2005). This transition intensity defines the frequency at which Equation (4.2) occurs, given that there are $N_i$ molecules extant in state $S_i$.

Let $N_i(t)$ be the number of ants in state $S_i$ at time $t$. Since the number of ants in the system is conserved over time, we have that

$$\sum_{i \in S} N_i(t) = \sum_{i \in S} N_i(0), \text{ for all } t > 0. \tag{4.3}$$

We will use this equation in Section 4.5 to determine the number of detached ants at any given time as

$$N_D(t) = \sum_{i \in S} N_i(0) - (N_B(t) + N_F(t)). \tag{4.4}$$

We stated in Section 4.2 that the load was conveyed in an approximately straight path. Consequently, our model for the load dynamics will be one dimensional, characterized by the load position $x_L$ and load velocity $v_L = \dot{x}_L$. From our observations, it was reasonable to assume that each *Front* ant pulled the load with the same pulling force $F_p$, and every attached ant lifted the load with the lifting force $F_L$. Moreover, since the load was conveyed

at approximately constant velocity, we assumed a proportional velocity regulation for the pulling force,

$$F_p = K(v_L^d - v_L(t)) \tag{4.5}$$

Here, $K$ is the proportionality constant of the load velocity $v_L(t)$ that is being regulated towards a desired velocity $v_L^d$.

At time instant $t$, the load is being lifted by a total of $N_B(t) + N_F(t)$ ants, each lifting with force $F_L$, while the load presses downwards with a weight of $m_L g$. Assuming that the load is in static equilibrium in the vertical direction, it experiences an upward normal force of

$$F_n(t) = m_L g - (N_B(t) + N_F(t))F_L. \tag{4.6}$$

At time $t$, the load is pulled with force $F_p$ by $N_F(t)$ ants, and is held back by a frictional force amounting to $\mu F_n(t)$. Then the net force experienced by the load in the $x$-direction is

$$F(t) = N_F(t)F_p - \mu F_n(t) \tag{4.7}$$

**Formulating the SHS** We now define the state vector $\mathbf{x}$ of our SHS (Hespanha and Singh, 2005; Hu *et al.*, 2000), combining the behavioral and dynamical components as

$$\mathbf{x} := \begin{bmatrix} N_F & N_B & N_D & x_L & v_L \end{bmatrix}^T. \tag{4.8}$$

The *flow* of the SHS is given by

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 0 & v_L & F/m_L \end{bmatrix}^T, \tag{4.9}$$

where $F$ is defined by Equation (4.7).

In addition, the SHS is governed by the six transitions of the form Equation (4.2), with the reaction $S_i \to S_j$ decrementing $N_i$ and incrementing $N_j$. As such, this reaction gives rise to the *reset map* $(N_i, N_j) \to (N_i - 1, N_j + 1)$ and has the transition intensity $\lambda_{ij}$. Since the transition intensities, reset maps and flow are all polynomial functions of the state variables, our SHS is a *polynomial* Stochastic Hybrid System, or pSHS. The discrete behavioural components $N_i(t)$ drive the changes in the continuous variables $x_L(t)$ and $v_L(t)$. Equation (4.9) describes the time evolution of the SHS, considering its components to be continuous variables, with a set of Ordinary Differential Equations (ODEs). Unlike Equation (4.2), it abstracts away individual transitions, describing instead the populations of ants in each state. As such, we call it the *macroscopic* (or macro, in short) model of the SHS.

## 4.4 Computing Moments of the State Vector

The *extended generator operator L* of an SHS may be used to derive the time evolution of the moments of its state vector (Hespanha and Singh, 2005; Hespanha, 2004) as stated in Section 3.2. For our SHS, we may compute the extended generator $L$ to be

$$L\psi(\vec{x}) \triangleq \frac{\partial \psi}{\partial x_L}\dot{x}_L + \frac{\partial \psi}{\partial v_L}\dot{v}_L + \sum_{\substack{i,j \in \{F,B,D\} \\ i \neq j}} \left(\psi(\phi_{ij}(\vec{x})) - \psi(\vec{x})\right) r_{ij} N_i.$$

By successively setting $\psi := N_i$ (for getting the mean number of ants in each state), $\psi := x_L$ (for load position), and finally $\psi := v_L$ together with the approximation that $N_F$ and $v_L$ are

uncorrelated (see Figure 4.3), we obtain

$$\frac{d}{dt}\mathrm{E}(N_i) = \sum_{\substack{j\in\{F,B,D\}\\ j\neq i}} \left(r_{ji}\mathrm{E}(N_j) - r_{ij}\mathrm{E}(N_i)\right),$$

$$\frac{d}{dt}\mathrm{E}(x_L) = \mathrm{E}(v_L), \tag{4.10a}$$

and

$$\frac{d}{dt}\mathrm{E}(v_L) = c_g + c_F\mathrm{E}(N_F) + c_B\mathrm{E}(N_B) + c_{Fv}\mathrm{E}(N_F)\mathrm{E}(v_L) \tag{4.10b}$$

where

$$c_g \triangleq -\mu g, \qquad c_F \triangleq (Kv_L^d + \mu F_l)/m_L \tag{4.11}$$

$$c_B \triangleq \mu F_l/m_L, \qquad c_{Fv} \triangleq K/m_L \tag{4.12}$$

.

## 4.5 Estimating SHS Parameters

Now we detail the estimation of the transition rates $r_{ij}$ and the parameters $K$ and $v_L^d$ of the proportional controller in Equation (4.5). We compute the means of the experimental data, and find the parameters that best fit the moment equations Equation (4.10). The experimental data gives us the empirical values for $N_F, N_B$ and $x_L$ at each $5s$ interval of a $145s$ time span. The values of $N_D$ over each $5s$ interval were computed using Equation (4.4); also, $v_L$ was computed from the values of $x_L$. Next, we computed the across-trial means of each of the components of $\mathbf{x}$ over each time interval.

A *Weighted Least Squares* (WLS) optimization approach was used to fit the theoretical moments in Equation (4.10) to the experimental means thus obtained, with the squared error of each variable weighted by the sample variance of its dataset. Specifically, the WLS procedure minimized the difference between the sum of weighted squared errors between each theoretical and empirical mean, where each weight was the sample variance of the corresponding data set. This optimizer was implemented in MATLAB, using the `fmincon` tool for active-set optimization, and the theoretical mean field trajectories were integrated using `ode15`.

$$r_{DB} = 0.0197\,\mathrm{s}^{-1}, \qquad r_{BD} = 0.0205\,\mathrm{s}^{-1},$$
$$r_{DF} = 0, \qquad r_{FD} = 0,$$
$$r_{BF} = 0.0301\,\mathrm{s}^{-1}, \qquad r_{FB} = 0.0184\,\mathrm{s}^{-1},$$

and the gain and velocity set point parameters

$$K = 0.0035\,\mathrm{N/(cm/s)}, \qquad v_L^d = 0.3185\,\mathrm{cm/s}.$$

Table 4.1 shows the standard errors in the fit.

### *4.5.1 Note on Model Fitting Process*

The WLS fitting process does not employ a more sophisticated model validation process such as cross-validation or bootstrapping as a safeguard against overfitting. This is because our goal was to extract a model from data on ant-based CT and subsequently *apply* this

**Table 4.1:** Standard Error in WLS Fitting

| State Variable | Standard Error |
|---|---|
| $N_F$ | 0.0283 |
| $N_B$ | 0.0082 |
| $N_D$ | 0.0408 |
| $x_L$ | 0.4147 cm |
| $v_L$ | $0.0029\ \text{cm} \cdot \text{s}^{-1}$ |

model to design control policies for multi-robot CT, rather than develop an accurate model of the former. Consequently, our validation process performed in (Wilson *et al.*, 2014) involved comparing the CT of ants with that robots. Three distinct robotic CT schemes are developed therein, which mimic ant-based CT to a different degree.

## 4.6 Inferences from Estimated Parameters

Since $r_{DB} > r_{DF} = 0$ , we infer that *Detached* ants are more likely to attach to the *Back* of the load rather than the *Front*, possibly because of the following reasons: The front of the load is already filled with pulling ants, making it easier for a *Detached* ant to attach to *Back*. Also, it may be easier for a *Detached* ant to attach to *Back* and walk forward to the nest rather than attaching to *Front* to walk backwards. The relatively high value of the *Back*-to-*Front* transition rate $r_{BF}$ may indicate the ants' willingness to adopt the less preferred posture of walking backwards when faced with transporting a heavy load. As shown in Figure 4.4, the predictions of the model agree quite well with the means of the experimental data.

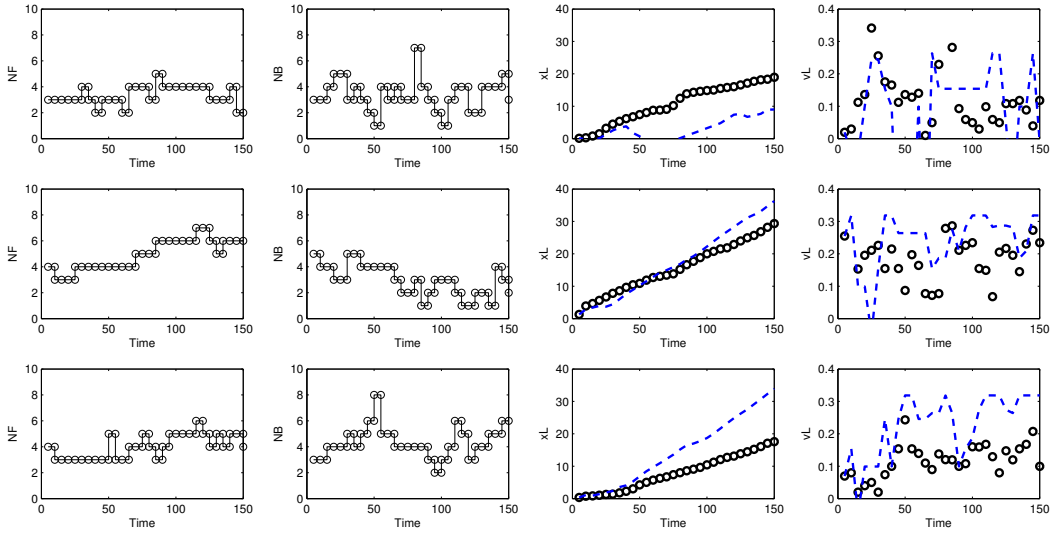



**Fig. 4.4:** Observed numbers of ants in states *Front* (first column) and *Back* (second column) over time and observed (circles) and predicted (dashes) load position (third column) and velocity (fourth column) over time for three selected experimental trials (one per row).

In Figure 4.5, the first row shows the mean number of ants in each behavioral state $S_i$ computed from averaging the data, compared against its counterpart predicted by the mean-field model. The second row shows analogous comparisons for the dynamical variables $x_L$ and $v_L$.

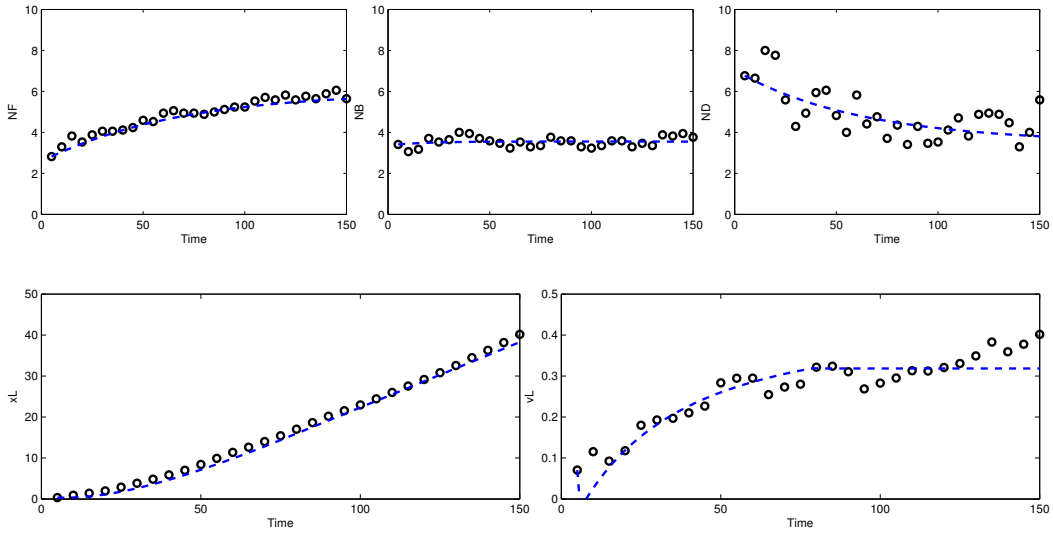



**Fig. 4.5:** Observed (circles) and predicted (dashes) time evolution of the mean numbers of ants in each state (top row) and the mean load position and velocity (bottom row).

Chapter 5

# STOCHASTIC CONTROL POLICIES FOR THE COVERAGE OF MULTIPLE LOAD BOUNDARIES

**Sources**: (Pavlic *et al.*, 2013a, 2015; Wilson *et al.*, 2014)

Now we will address the problem of allocating robots to multiple load boundaries, with the objective of achieving a target allocation of robots to each boundary. Loads may be of different predefined types, with each type requiring a distinct target allocation. Our approach uses a macroscopic model of robot population dynamics to design a stochastic control policy that achieves the desired allocation to each boundary type. This policy relies only on local information, yet provides theoretical guarantees on the allocation achieved.

In our environment, robots randomly walk on the plane, avoid collisions with each other, and bind to disk-shaped loads. In the course of binding, they may require other robots to unbind from (i.e. leave) the boundary. We will validate this policy and demonstrate through simulations that it is robust to changes in the number of robots and loads. Later, we will modify our control policy to avoid the need for inter-robot communication. We will also explain how our controller can be used to mimic the ant-based collective transport of Chapter 4.

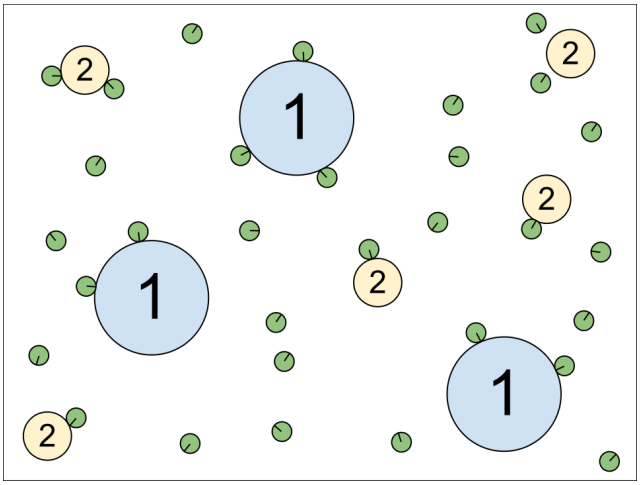



**Fig. 5.1:** Example scenario with two types of disk-shaped loads, labeled 1 and 2, taken from (Pavlic *et al.*, 2015). The unlabeled circles are robots that are allocating themselves to the load boundaries.

## 5.1 Problem Statement

An example scenario is shown in Figure 5.1, which displays a set of homogeneous disk-shaped green robots together with two types of loads, distinguished by labels 1 and 2. For simplicity, we will consider only disk-shaped loads. While disks are distinguished by size in Figure 5.1, this distinction could be based on other properties in general. Suppose that we need a mean allocation of 3 robots per type-1 load, and 5 robots per type-2 load. The robots have no prior information about the loads; instead, each robot encounters loads in the course of randomly walking in the plane, and can sense and communicate only within a pre-specified radius.

*Problem:* Given the following information: robot radius, radius of each load type, and target allocation to each load type, design and validate a robot controller that provably achieves the desired allocation at steady state.

## 5.2 Design of Stochastic Controller

Figure 5.2 displays the finite state machine of a stochastic controller that drives robot behavior. There are three types of robot behaviors:

1. Robots start in the *Search* state by executing a *correlated random walk* (CRW) in the plane from their starting positions. This means that each robot travels straight ahead for a single unit of time, then turns at a random angle, and repeats this behavior indefinitely. A CRW is thus completely specified by its step length and maximum turning angle. Thus far, robots have not attached or *bound* to any load, and are called *free* or *unbound*.

2. **Robot-Load interactions**: On encountering a load, the robot is capable of identifying the load type. It may then choose to bind to the load with probability $p_b(type)$, which depends on the load type. For conciseness, we will write only $p_b$ from now. Otherwise, it will continue its CRW, remaining unbound.

3. **Robot-Robot interactions**: When robot $\mathscr{R}_1$ encounters $\mathscr{R}_2$ in its neighborhood, each sends out a message to the other, indicating whether it is bound or unbound. If both are unbound, then each executes a collision-avoiding maneuver. Otherwise, one (say $\mathscr{R}_2$) is bound, and the other (say $\mathscr{R}_1$) is unbound. Then with probability $p_u(type)$, which will just be called $p_u$ from now, $\mathscr{R}_1$ may command $\mathscr{R}_2$ to unbind; else, $\mathscr{R}_1$ continues its CRW.

To make precise the design of the stochastic controller, we introduce the notions of *bound and unbound zones*. An *unbound zone* is defined to be a sector of the load, whose arc length

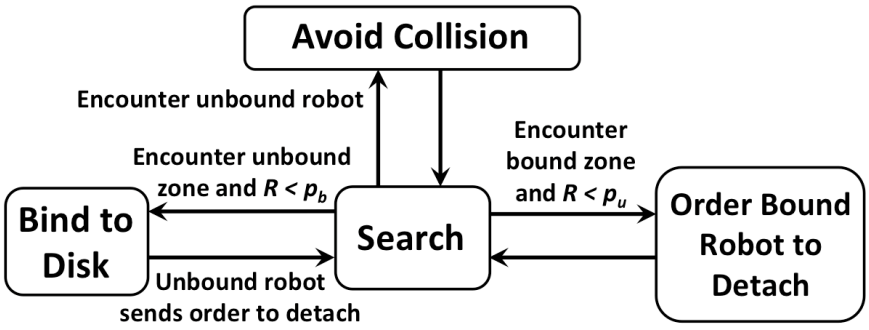



**Fig. 5.2:** Diagram of control flow, taken from (Pavlic *et al.*, 2013a). Robots randomly cycle through searching and encountering robots and unbound zones. On encountering a load, they probabilistically choose to bind to unbound zones or command bound robots to unbind. On encountering unbound robots, they avoid collisions. Probabilities can be implemented with a pseudo-random number $R \in \text{unif}(0,1)$.

is equal to the linear distance that a robot can occupy along the load boundary. A *bound zone* consists of a bound robot together with the sector of the load it is attached to (Wilson *et al.*, 2014).

## 5.3 Microscopic Model: Chemical Reaction Network

We may represent the robot controller by a CRN as follows. Supposing that there is only one load type, define three chemical species: $r$, representing free robots; $B$, representing bound zones; and $U$, representing unbound zones. Then the action of the controller may be captured by the reactions:

$$r + U \xrightarrow{p_b e_u} B \tag{5.1a}$$

$$r + B \xrightarrow{p_u e_b} U + 2r \tag{5.1b}$$

Equation (5.1a) specifies that a free robot will bind to an unbound zone with probability $p_b$; likewise, Equation (5.1b) tells us that a free robot will command a bound robot to unbind with probability $p_u$. The constants $p_b e_u$ and $p_u e_b$ are called *mass-action rate constants*, as

in chemical kinetics. The quantities $e_u$ and $e_b$ are called *encounter rates*, with $e_u$ (resp. $e_b$) defined as the probability rate that a free robot will encounter an unbound (resp. bound) zone during its search. Both these encounter rates are environmental parameters, and are outside the designer's control. On the other hand, the probabilities $p_u$ and $p_b$ are to be set by the designer.

A CRN that consists of a pair of forward and reverse reactions will achieve a desired equilibrium concentration of reactants and products. While Equation (5.1a) is a forward reaction, Equation (5.1b) is not quite a reverse reaction, though it has the same effect. The effect of the free robot $r$ in Equation (5.1b) is to free the bound robot, and thereby increase the number of unbound zones. This robot behaves like an *enzyme* which accelerates the production of unbound zones, while remaining unaffected by the reaction. While the encounter rates $e_u$ and $e_b$ may change with the number of bound and unbound zones, the reaction pair above allows us to define the probabilities $p_u$ and $p_b$ in terms of the *ratio* $\frac{e_b}{e_u}$. As we show in (Pavlic *et al.*, 2013a), this ratio varies predictably with parameters of the robot motion primitives, but is robust to a wide range of environmental variations. This property allows us to select a $p_u$ and $p_b$ that consistently drive the system to a target allocation in different types of environments.

## 5.4 Macroscopic Model: Concentration Fields

Using the theory of mass-action kinetics, we find that the CRN in Equation (5.1a), Equation (5.1b) corresponds to a macroscopic model consisting of the following set of first-order

ODEs:

$$\begin{bmatrix} \dot{r} \\ \dot{U} \\ \dot{B} \end{bmatrix} = (p_u e_b r B - p_b e_u r U) \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}, \tag{5.2}$$

where $r$, $U$, and $B$ denote the concentrations of their associated species. This system can be solved completely when the initial conditions $\begin{bmatrix} r_0 & U_0 & B_0 \end{bmatrix}^T$ are prescribed.

We will now find the equilibrium concentration $\begin{bmatrix} r^* & U^* & B^* \end{bmatrix}^T$ of the species. A trivial equilibrium is reached when $r^* = 0$, i.e. when there are no free robots left, and all are bound. If we assume that $r^* > 0$, then we have an additional *nontrivial* equilibrium at which $B^*$ and $U^*$ obey the relation:

$$\frac{B^*}{B^* + U^*} = \frac{p_b e_u}{p_b e_u + p_u e_b} = \frac{\frac{p_b}{p_u}}{\frac{p_b}{p_u} + \frac{e_b}{e_u}}. \tag{5.3}$$

Further, noting that $\dot{U} = -\dot{B} = \dot{r}$, we need $B^* + U^* = B_0 + U_0$. We now have two equations in two unknowns, which may be solved uniquely for $U^*$ and $B^*$. We then find that the number of free robots is driven to the equilibrium (Pavlic *et al.*, 2013a)

$$r^* = (B_0 + r_0)\frac{p_u e_b}{p_b e_u + p_u e_b} - (U_0 - r_0)\frac{p_b e_u}{p_b e_u + p_u e_b}, \tag{5.4}$$

Further since $\dot{U} = -\dot{B} = -\dot{r}$, the sum of the bound and unbound zones is conserved, yielding

$$B^* + U^* = B_0 + U_0 \tag{5.5}$$

## 5.5 Stability of the Nontrivial Equilibrium

Noting from Equation (5.2) that $\dot{r} = \dot{U} = -\dot{B}$, we can rewrite the third order system in it as a first-order ODE in terms of $r$ alone:

$$\begin{aligned}\dot{r} &= p_u e_b r(\overbrace{B_0 - (r - r_0)}^{B}) - p_b e_u r(\overbrace{U_0 + (r - r_0)}^{U}) \\ &= ((B_0 + r_0)p_u e_b - (U_0 - r_0)p_b e_u)r - (p_b e_u + p_u e_b)r^2.\end{aligned} \tag{5.6}$$

Assuming that mass-action rate constants are nonzero, the non-trivial equilibrium of Equation (5.6) is at $r = r^* > 0$, where

$$r^* = (B_0 + r_0)\frac{p_u e_b}{p_b e_u + p_u e_b} - (U_0 - r_0)\frac{p_b e_u}{p_b e_u + p_u e_b}.$$

From Equation (5.3) and Equation (5.5), we have

$$r^* = (B_0 + r_0)\left(1 - \frac{B^*}{B_0 + U_0}\right) - (U_0 - r_0)\frac{B^*}{B_0 + U_0} \tag{5.7}$$

which is positive and asymptotically stable so long as $B_0 + r_0 > B^*$. More specifically, the system in Equation (5.2) has an asymptotically stable equilibrium described by

$$(r, B, U) = \begin{cases} (r^*, B^*, U^*) & \text{if } r_0 + B_0 > B^*, \\ (0, U_0 - r_0, B_0 + r_0) & \text{otherwise.} \end{cases} \tag{5.8}$$

So the CRN in Equation (5.1) is driven by the imbalance between fluxes to and from bound and unbound zones; it comes to rest when enough free robots are converted into bound zones to restore flux balance or when the pool of free robots is totally depleted.

### 5.5.1 Asymptotic Stability of the Nontrivial Equilibrium

Now we will prove that the equilibrium in Equation (5.8) is asymptotically stable. We find that $\dot{r}$ is stationary at the point

$$r = \frac{2(p_b e_u + p_u e_b)}{((B_0 + r_0)p_u e_b - (U_0 - r_0)p_b e_u} = r^*/2, \tag{5.9}$$

where the last equality follows from Equation (5.7). From Equation (5.6), we find that $\dot{r}$ is positive at the stationary point, and $\dot{r}$ is negative for $r > r^*$. Further, the stationary point $r^*/2$ is a maximum, for $\ddot{r}|_{r=r^*/2}$ is negative. These observations lead us to the phase plot shown in Figure 5.3. The flow lines converge to the nontrivial equilibrium $r = r^*$, making this a stable equilibrium. Finally, since $r^*$ is strictly positive, this equilibrium is asymptotically stable. Since $\dot{r} = \dot{U} = -\dot{B}$, similar observations apply to $B^*$ and $U^*$, making the equilibrium $(r^*, U^*, B^*)$ asymptotically stable.

## 5.6 Reduced-Order Model and its Applications

Let $r_0$ be the initial number of free robots. Assuming that no robots are initially bound, then $r_0$ is the total size of the swarm. For a large robotic swarm with $r_0 \gg B^*$, the concentration $[r]$ will essentially be constant. That is, while the concentrations $[B]$ and $[U]$ will move opposite to each other in relatively large swings, the fractional change in $[r]$ will be negligible (Pavlic *et al.*, 2013b). Consequently, the bimolecular reactions in Equation (5.1) can be viewed as the reversible unimolecular reaction

$$U \underset{p_u e_b r_0}{\overset{p_b e_u r_0}{\rightleftharpoons}} B, \tag{5.10}$$

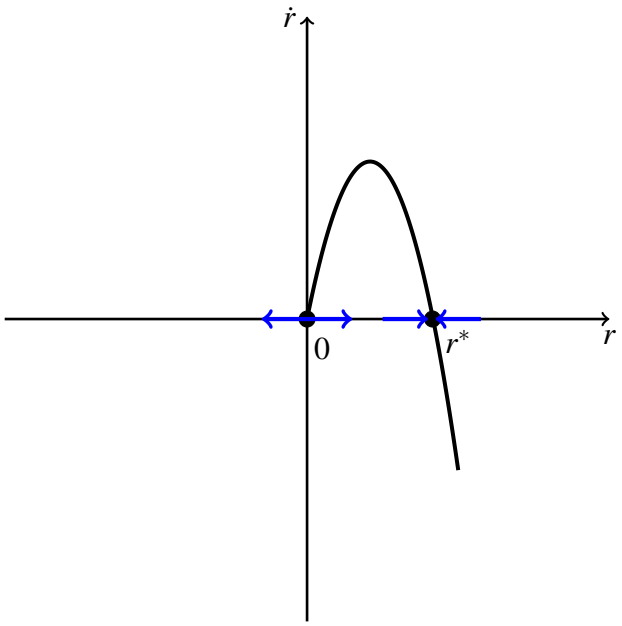



**Fig. 5.3:** Phase plot of $r$ vs $\dot{r}$ from Equation (5.6), showing equilibria at $r = 0$ (trivial,unstable equilibrium) and $r = r^*$ (nontrivial, stable equilibrium). Flow arrows are drawn in blue.

which introduces the swarm size $r_0$ as a scalar on both reaction rate constants. This unimolecular approximation will also hold close to the equilibrium of Equation (5.1). Thus, it also represents how the system returns to equilibrium after small perturbations, and its equilibrium matches the one predicted for the full bimolecular model. Moreover, the mean-field dynamics of this unimolecular CRN is a second-order linear time-invariant ODE, which allows for exponential characterizations of its convergence rate.

### *5.6.1 Convergence Rate and Variance*

Unimolecular reversible reactions have been used in much of the stochastic task-allocation literature. For these systems (Odhner and Asada, 2010, Section 3.2), the convergence rate is

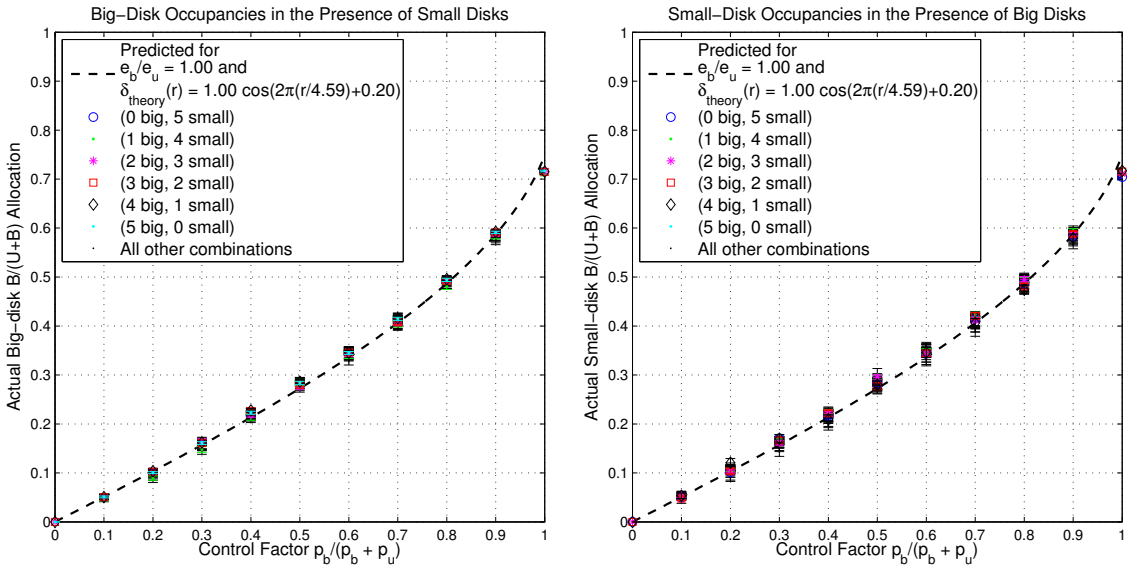
independent of the equilibrium distribution. For this system in particular, the time constant

$$\tau \triangleq \frac{1}{(p_b e_u + p_u e_b) r_0}. \tag{5.11}$$

So increasing the swarm size $r_0$ has no effect on the equilibrium distribution; however, it increases the speed of the system. Moreover, by similar reasoning for unimolecular reversible reactions (Odhner and Asada, 2010, Section 3.2), the allocation variance is independent of the convergence rate. So maximizing the convergence rate of the system has no cost in terms of variance. Consequently, arbitrarily large swarms can be used to quickly achieve and maintain the equilibrium allocation with very little performance cost. The larger the swarm, the faster and more linear the system behaves. As shown in Figure 5.4, variance decreases with the number of zones, which is consistent with unimolecular arguments about control of variance (Odhner and Asada, 2010). So the variance of the system will be improved by using robots which are relatively small with respect to the boundaries they allocate to. By using small robots, the number of zones necessarily increases, and the variance in allocation decreases accordingly.

### *5.6.2 Optimal Choice of Control Policy*

We have discussed how increasing swarm size and number of zones has an effect on convergence rate and allocation variance without affecting the equilibrium mean allocation ratio. Furthermore, the control policy in Equation (5.14) includes one degree of freedom over which $(p_b, p_u)$ pairs can be chosen to optimize some feature of the system with no effect on mean allocation ratio. By Equation (5.11), the convergence rate of the system can be maximized by making the sum $p_b + p_u$ as large as possible. Thus, for the fastest convergence to a desired $B^*/(U^* + B^*)$ equilibrium occupancy, the controls $p_b$ and $p_u$ should




(a) Big-disk allocations

(b) Small-disk allocations

**Fig. 5.4:** Effect of varying disk mixture. Ten trials were generated for each experimental treatment (i.e., number of big disks, number of small disks, big allocation ratio, and small allocation ratio); the big-disk type is twice as large as the small-disk type. The averages shown for each big-disk allocation in (a) are taken across the pool of 110 trials that include all eleven small-disk allocation ratios for the corresponding big-disk allocation ratio and big–small mixture (and similar for the statistics in (b)). Error bars show $\pm 1$ standard error of the mean (SEM). The broken line in the plot shows the expected allocation curve from theory.

be chosen so that

$$(p_b, p_u) = \begin{cases} \left( \frac{e_b}{e_u} \frac{B^*}{U^*} \frac{(1+\delta)}{1-\delta B^*/U^*}, 1 \right) & \text{if } e_b B^* < e_u U^* \frac{1-\delta B^*/U^*}{1+\delta}, \\ \left( 1, \frac{e_u}{e_b} \frac{U^*}{B^*} \frac{1-\delta B^*/U^*}{(1+\delta)} \right) & \text{otherwise.} \end{cases} \tag{5.12}$$

However, in an implementation of these strategies on real robots subject to mechanical fatigue or appreciable task-switching times, there may be some constraint which limits the $p_b + p_u$ sum.

## 5.7 Correction for Spatial Effects

In principle, robots can fill the boundary completely, leaving no inter-robot space; however, this situation rarely arises in practice, for our stochastic control policy may place two robots close enough to each other that it is impossible for a third one to attach to the space in between them. Mathematically, the maximum value of the ratio $\frac{B}{B+U}$ is unity in theory, which is hard to achieve in practice. Define $r$ to be the mean size of the inter-robot space between two bound zones. In (Pavlic *et al.*, 2013a), we empirically determine the functional form of $r$ and show that the spatially-corrected equilibrium allocation is given by

$$\frac{B^*}{U^*+B^*} = \frac{B^*}{(U^*-\delta B^*)+(1+\delta)B^*} = \frac{1}{1+\delta}\overbrace{\frac{\frac{p_b}{p_u}}{\frac{p_b}{p_u}+\frac{e_b}{e_u}}}^{\text{Idealized allocation}}. \tag{5.13}$$

## 5.8 Control Synthesis for Multiple Disk Types

For initial number of free robots $r_0$ (i.e., roughly the size of the swarm), if $r_0 > B^*$, then the swarm allocation system for one disk type will converge to the equilibrium described by Equation (5.13). Thus, a $(p_b, p_u)$ control policy for the single-type case can be synthesized using the rule

$$\frac{p_b}{p_u} = \frac{e_b}{e_u}\frac{B^*}{U^*}\frac{(1+\delta)}{1-\delta\frac{B^*}{U^*}} \tag{5.14}$$

where $B^*/U^*$ is the desired bound–unbound ratio of zones at equilibrium. For example, if robots at equilibrium are to occupy 60% of a disk boundary on average, then the desired ratio $B^*/U^* = 0.6/(1-0.6) = 1.5$, and Equation (5.14) can be applied to generate the required $p_b/p_u$ ratio for any given $e_b/e_u$ ratio. For any given desired allocation, there is a continuum of corresponding $(p_b, p_u)$ pairs. Consequently, as we demonstrate in Sec-

tion 5.6.2, the absolute values of $p_b$ and $p_u$ can be chosen to optimize some other criteria so long as the $p_b/p_u$ ratio satisfies Equation (5.14).

Under the assumption that a robot can detect the difference between disk types, the robotic swarm can be configured to achieve a different equilibrium allocation for each disk type. If one additional disk type is added to the system, the CRN from Equation (5.1) can simply be augmented with the two additional reactions

$$r + U' \xrightarrow{p'_b e'_u} B' \tag{5.15a}$$

$$r + B' \xrightarrow{p'_u e'_b} U' + 2r \tag{5.15b}$$

where $U'$ and $B'$ represent the unbound and bound zones on the new disk type, and $p'_b e'_u$ and $p'_u e'_b$ represent the mass-action rates of the new reactions. So long as $r_0 > B^* + B'^*$, then both pairs of reactions in Equation (5.1) and Equation (5.15) will meet the sufficient conditions for convergence to equilibria described by Equation (5.13). Thus, the $(p_b, p_u)$ pair for the first type will be independent of the introduction of the second type, and the $(p'_b, p'_u)$ pair for the second type can be designed independently of the control pair for the first type. Moreover, so long as the robotic swarm has a sufficiently large size $r_0$, any number of disk types can be considered, and the design for each type will be independent of the design for every other type.

The catalytic reverse reactions in Equation (5.1b) and (5.15b) allow for the multi-type decomposition described above. In general, regardless of the number of disk types, adding more disks to a finite volume reduces the free space available for movement. Consequently, the introduction of new disks increases the encounter rates between robots and disks. If, instead of a catalytic reverse reaction, a traditional timer-based reverse reaction was imple-

mented, the mean binding time would have to be reduced on every robot to compensate for the increased robot density. That is, the increased density would lead to a higher binding event rate, and so the unbinding event rate would have to be similarly increased to maintain the desired equilibrium allocations. As discussed in Section 5.3, the catalytic reverse reactions allow for the control policy in Equation (5.14) to be independent of density. Therefore, so long as the robotic swarm is sufficiently large, additional disk types can be added with no impact on existing control designs.

## 5.9 Validating the Stochastic Controller

To validate our analytical results above, we ran a large number of agent-based simulations using NetLogo (Wilensky, 1999). We first validated our formula Equation (5.13) for spatially-corrected equilibrium allocations. We also demonstrated the robustness of our approach to variations in environmental parameters, as shown in Figure 5.5. Figure 5.6 plots robot allocations over time in a simulation with 300 robots, demonstrating that our designed $(p_b, p_u)$ results in the target steady-state allocation.

## 5.10 Simulating Ant-based Collective Transport

Drawing from (Wilson *et al.*, 2014), we will briefly outline how the stochastic controller above can be modified to mimic the SHS-based collective transport model of Chapter 4. As shown there, we will divide every load into *Front* and *Back* halves; accordingly, we now have four probabilities $p_{uB}, p_{bB}, p_{uF}$, and $p_{bF}$ that govern binding and unbinding at each half. The load dynamical model is carried over from Section 4.3. The SHS model

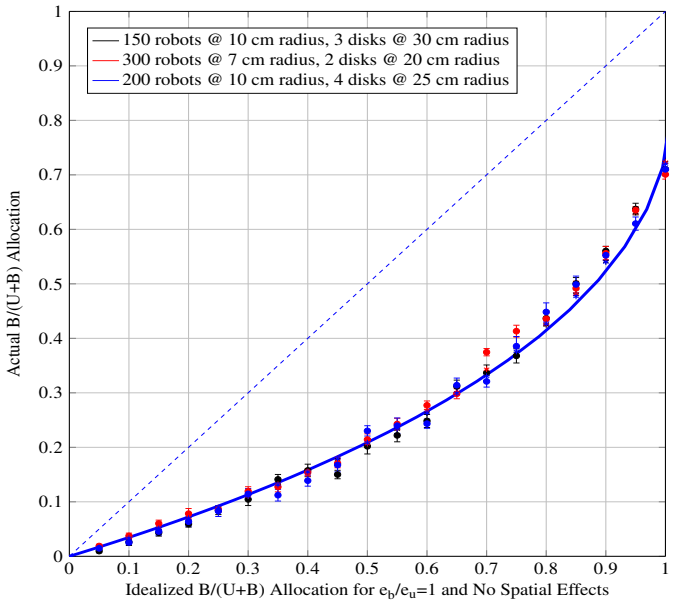



**Fig. 5.5:** Effect of varying environmental parameters on robot allocations (Pavlic *et al.*, 2013a). Ten trials were generated for each disk size, and the average across the trials are shown with error bars indicating $\pm 1$ standard error of the mean. A dashed line of unity slope is shown for reference. The solid line represents the predicted curve based on the robot collision-avoidance distance $a$, which is non-zero for these cases.

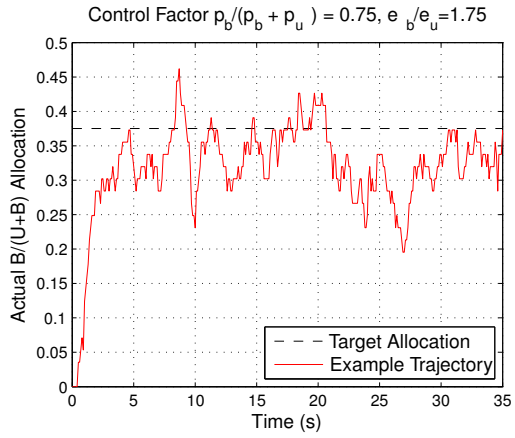
converges exponentially to the following mean steady-state ant populations:

$$[Front]^* \approx 5.78 \text{ ants} \quad \text{and} \quad [Back]^* \approx 3.54 \text{ ants} \tag{5.16}$$

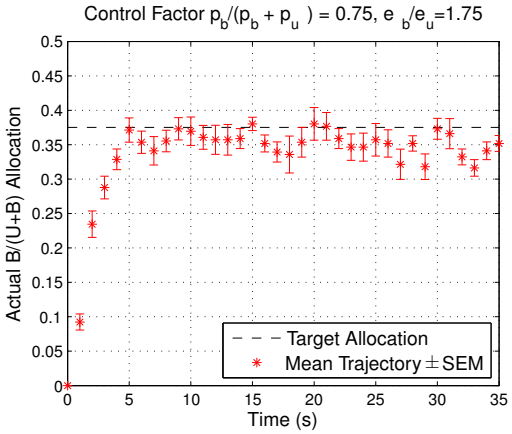
$$\text{with time constant} \quad \tau \approx 52.08\,\text{s}. \tag{5.17}$$

In (Wilson *et al.*, 2014), we develop and validate three methods for computing $p_{uB}, p_{bB}, p_{uF}$, and $p_{bF}$ in order to drive a robotic swarm to mimic different aspects of ant-based collective transport:

1. *Equilibrium Population Matching Method (EPMM)*, which reproduces the mean steady-state *Front* and *Back* ant populations, but does not necessarily replicate the transient dynamics of the ant populations.




(a) Single trajectory




(b) 10-simulation average trajectory

**Fig. 5.6:** Example simulated trajectories of robot allocation (Pavlic *et al.*, 2015). In (a), a single execution of a simulation of 300 robots allocating to 3 disks is shown. In (b), the average trajectory is shown across 10 simulations. The $(p_b, p_u)$ policy was chosen to achieve a mean $B/(B+U)$ allocation of 37.5%, shown as a dashed horizontal line.

2. *Transient Matching Method (TMM)*, which reproduces the transient dynamics and equilibrium populations, but requires knowledge of absolute encounter rates for its implementation.

3. *Rate Matching Method (RMM)*, which reproduces the transition rates (and hence, fluxes) between states, but not necessarily the equilibrium allocations and system convergence rate.

## 5.11 Comparison with Existing Work

### *5.11.1 Comparison with (Konur* et al.*, 2012)*

*Summary of (Konur* et al.*, 2012)*: The paper develops a controller for a swarm of foraging robots. The goal of the controller is to make the robots procure enough food for themselves to enable their long-term survival. The individual robot controller is driven by a 5-state probabilistic FSM (PFSM). However, formally verifying the behavior of even small swarms takes up excessive memory and time due to state space explosion. To circumvent the difficulty of verification, the paper develops a macroscopic model that models the entire swarm as a single PFSM. The properties of the macro model was then verified for several test cases.

The *designable parameters* of the model are the probabilities that (1) a robot will forage (instead of remaining hungry) $\gamma_f$ and (2) it will grab food that has foraged (instead of disacrding it) $\gamma_g$. The long-term survival of the swarm depends on these probabilities, as

well as the energy $E_d$ delivered by a single food item. The goal is to determine whether the swarm will reach a prescribed energy level $R$ within time $T$, given $(E_d, \gamma_f, \gamma_g)$.

I will call this work the *Foraging* scenario and our work (Pavlic *et al.*, 2013a, 2015; Wilson *et al.*, 2014) the *Robozyme* scenario.

**Similarities** : In both scenarios, the micro model consists of a PFSM, whose probabilities / rates can be designed to achieve the system's goal. Due to the complexity of analyzing the micro model directly, a simpler macro model is devised. The macro model "smooths out" or "averages" the behavior of individual robots, making it easier to analyse global behavior. The macro model in Foraging keeps track of robot counts in each state; that in Robozyme keeps track of the spatial density of robots (or "concentrations" of species) over time.

Beyond these surface similarities, the two scenarios are substantially different.

**Provable properties**: Note from Table 5.1 that the nature of guarantees provided by Robozyme and Foraging are quite distinct.

*Robozyme* offers probabilistic guarantees on : (1) the mean number of robots alloted to each load type in steady states , and (2) the time it takes to converge to this steady state allocation. Formal model checking is impossible here, due to the difficulty in determining two spatial effects: *encounter rates*, or the probability that a robot executing CRW will encounter a load and *boundary spatial effects*, which prevent a boundary from getting completely filled with robots. As a result, the Robozyme model is validated through MC simulations.

*Foraging* offers *deterministic guarantees* on the energy level $R$ reached by the swarm within a time bound $T$ of the following form: Given fixed values of the triple $(E_d, \gamma_f, \gamma_g)$, and the

**Table 5.1:** Comparison between Robozyme and Foraging Models

| **Aspect** | **Robozyme** | **Foraging** |
|---|---|---|
| *Domain* | Boundary coverage: allocating to loads | Fetching food to nest |
| *Method of Verification* | Monte Carlo(MC) Simulation | Formal Model Checking, MC Simulations |
| *Goal* | Allocate to loads with prescribed *allocation per load type* | Forage with energy effiency |
| *Guarantees* | Steady-state allocation of desired number of robots to load type | Swarm reaches desired energy level for *all possible executions* |
| *Nature of Guarantee* | Probabilistic | Mainly deterministic |
| *Design parameters* | Binding and unbinding probabilities $p_b, p_u$ | Foraging and grabbing probabilities $\gamma_f, \gamma_g$ |
| *Robot capabilities* | Local Sensing and communication | No assumptions |
| *Robot locomotion* | Correlated Random Walk (CRW) | No assumptions |
| *Spatial effects* | Encounter rates, spatial correction | Not modeled |
| *Collision avoidance* | Not significant | Explicitly provided in model |

goal state $(R,T)$, the swarm will either attain or fail to attain the goal state, with *probability* 1.0. This guarantee be made through exhaustive simulation, hence requires probabilistic model checking. Such a guarantee that is unconditional on the path followed by the system is called a *safety guarantee*.

Monte Carlo Simulations are then run to *compare* their performance, in terms of time and memory, against that of model checking (even if they do not offer safety guarantees). The comparison proves that for several cases, verification runs faster than simulations, while offering safety guarantees. The excellent performance of verification is traced to the clever design of the macroscopic model. Thus, the probabilistic guarantees furnished by simulations are *supplementary* to the deterministic ones.

### *5.11.2 Comparison with Amoebot*

Here we compare our Robozyme model with Amoebot, a model for self organizing particles developed in (Derakhshandeh *et al.*, 2014b,a; Dolev *et al.*, 2013). Despite the superficial resemblance between the two models, they belong in disparate domains for their end goals are different. This disparity is reflected in the distinct assumptions they make about robot primitives and sensing capabilities. Nonetheless, they may form alternative approaches to the same problem, possibly with distinct pros, cons, and difficulty of implementation.

**Table 5.2:** Comparison between Robozyme and Amoebot Models

| **Aspect** | **Robozyme Model** | **Amoebot Model** |
|---|---|---|
| *Domain* | Swarm Robotics | Self-Organizing Modular Robotics |
| *Application Areas* | Construction, Drug Delivery, Distributed Sensing | Smart Materials, Autonomous Monitoring, Surgery |
| *Types of Problems Solved* | Boundary Allocation, Collective Transport | Shape Formation, Bridging, Infinite Coating |
| *Goal* | Allocate to boundary with provable mean allocation | Morph into goal shape, remaining connected always |
| *Environment* | Continuous Plane $\mathbb{R}^2$ | Discrete Tiling of $\mathbb{R}^2$ |
| *Robot capabilities* | Local sensing, no communication | Local communication, modest computational power |
| *Robot actions* | Random walk, Bind, catalyze unbind | Null, Turn, Expand, Contract, Divide, Kill |
| *Robot controller FSM* | Driven by binding & unbinding probabilities | Driven by flags of neighbours |

## 5.12 Conclusion

We developed a controller based on a probabilistic FSM to stochastically allocate robots to multiple payloads. We modeled the robot controller as a CRN and examined the effect of encounter rates and robot probabilities of binding and unbinding to loads. The macroscopic model consisted of an ODE whose equilibrium was shown to be attainable by a robotic swarm through the appropriate design of robots' binding and unbinding probabilities. This design was validated by simulations with NetLogo. The controller was also modified to mimic ant-based collective transport using three distinct methods.

Chapter 6

# MAINTAINING A WIRELESS NETWORK AROUND A BOUNDARY - PART I

**Sources**: (Kumar and Berman, 2014, 2016b,a)

## 6.1 Overview and Problem Statement

In this and the next chapter, we will answer Parts (3) and (4) of the main problem in Section 1.3.2. This chapter will address the subproblem of computing the properties of the communication graph $\mathscr{G}$ associated with a point robot team *uniformly randomly* covering a boundary. The methods developed here will be reused to solve extensions of this subproblem in Chapter 7. Throughout this chapter and Chapter 7, we will follow the notational conventions in Table 6.1.

As mentioned in Section 1.3, this chapter and the next provide precise analytical formulas for robot positions and inter-robot distances (called slacks) arising from SCS. Our results use a variety of tools: order statistics, computational geometry, and Poisson processes to derive these formulas. Further, we make use of counting complexity reductions to establish that certain SCS lead to a communication graph whose properties are intractable to compute. Our results on the spatial pdfs generated by collision-free(CF) attachments of finite robots follow in a logical progression from our treatment of point robots for which collisions are nonexistent.

**Table 6.1:** Typographic Conventions

| **Variable type** | **Font** | **Example** | **Use** |
|---|---|---|---|
| Real number | Lowercase | Position instance | $\mathtt{x}_i$ |
| Vector | Lowercase Bold | Position vector instance | $\mathbf{x}$ |
| Matrix | Blackboard Bold | Vertex matrix | $\mathbb{V}$ |
| Random variable | Uppercase | Position variable | $\mathtt{X}_1$ |
| Random vector | Uppercase Bold | Position vector | $\mathbf{X}$ |
| Density function | Lowercase | pdf of positions | $f_{\mathtt{X}_i}(\mathtt{x}_i)$ |
| Real space | Blackboard Bold | Real line | $\mathbb{R}$ |
| Geometric figure | Calligraphic | Slack simplex | $\mathscr{S}$ |

| **Variable** | **Meaning** |
|---|---|
| $(\mathtt{x}, \mathtt{x}^T)$ | (Column, Row) vector |
| $\mathtt{x}_n$ | Column of $n$ entries |
| $\mathtt{x}_{a:b}$ | Entries $a \ldots b$ of $\mathtt{x}$ |
| $(\mathbf{0}, \mathbf{1})$ | Column of (zeros, ones) |
| $\mathbf{1}\mathbf{1}^T$ | Matrix of ones |

### 6.1.1 Models of Boundary Coverage

We consider a team of robots, $\{\mathscr{R}_i\}_{i\in[1\ldots n]}$ in a bounded environment $\mathscr{E}$. Robots are provided only with their (perfect) odometric readings and Wi-fi measurements, and a camera for detecting landmarks. Each robot is a disk of diameter $R$, and its Wi-fi has a coverage radius of $d$. They have no knowledge of their global positions or other means to localize. In the environment is placed a boundary in the form of a thin line, called the *Boundary* $\mathscr{B}$ which is colored black, distinctively from the rest of the environment. One endpoint of $\mathscr{B}$ is colored white.

Since the main thrust of this paper lies in the randomized analysis of the network created by the robots, we will make the following simplifying assumptions. All robots are synchronized in time, with respect to a global clock. No robot fails in the course of its execution. Adapting to failures, and determining the success of boundary coverage in rugged environments where Wi-fi may fail are issues to be addressed in our future work described in Chapter 8.

To begin with, we consider point robots, for which $R = 0$ and thus the issue of inter-robot collisions does not arise. Let $n$ of them attach to $\mathscr{B}$ at a time instant $t \in \mathbb{N}$ (Kumar and Berman, 2016b). Let the position of robot $i$ be $x_i$. Define the vector of *unordered* positions to be $\mathbf{x}(t) := [x_{1\ldots n}]^{\mathrm{T}}$. It will be convenient to make our computations if we sort this vector in nondecreasing order to get its permutation $\mathtt{x} = \left[\mathtt{x}_{1\ldots n}\right]^{\mathrm{T}}$, whose entry $\mathtt{x}_i$ is the position of the $i$-th robot from the left endpoint of $\mathscr{B}$, and not necessarily the position of $\mathscr{R}_i$. Since $\mathtt{x}_i$ forms the $i$-th smallest of the $n$ entries of $\mathtt{x}$, it is called the $i$-th *order statistic* of the positions (David and Nagaraja, 2003). We may think of $\mathbf{x}$ (and $\mathtt{x}$) as the realization of a PP

Section 3.7, so that $\mathbf{x}$ forms a point in $\mathscr{B}^n = \mathscr{B} \times \mathscr{B} \times \ldots \mathscr{B}$. Define the random variable (rv) associated with $\mathtt{x}_i$ to be $\mathtt{X}_i$, and place all these rv's in a vector $\mathtt{X}$ that defines the PP.

For convenience, we introduce two virtual robots $\mathtt{x}_0 = 0$ and $\mathtt{x}_{n+1} = s$ stationed at the endpoints of $\mathscr{B}$. Since connectivity depends with inter-robot distances, it helps to think of these distances directly rather than in terms of $\mathtt{x}$. Define the $i$-th *slack* $s_i$ to be the distance from $\mathtt{x}_i$ to $\mathtt{x}_{i+1}$, and the *slack vector* $\mathbf{s}$ to be $\mathbf{s}_{1:n+1} := \mathtt{x}_{1:n+1} - \mathtt{x}_{0:n}$ to be the vector comprised of all slacks. Analogous to the rv's (random variables) associated with positions, define the rv's $S_i$ and the vector $\mathbf{S}$. We may think of $\mathbf{s}$ as a point in $\mathscr{B}^{n+1}$.

Now we introduce the notion of connectivity by defining a *communication radius* or *connectivity threshold* $d \in [0, s]$. Two robots located at $\mathtt{x}_i$ and $\mathtt{x}_j$ are connected iff $|\mathtt{x}_i - \mathtt{x}_j| \leq d$. We model connectivity by a communication graph $\mathscr{G}(\mathtt{x})$, whose nodes are $\mathtt{x}_i$ (or $x_i$), and whose edges are formed by pairs of connected robots. Since each node is a geometric position, $\mathscr{G}(\mathtt{X})$ forms a *Geometric Graph*. We define a position vector $\mathtt{x}$ to be connected iff $\mathscr{G}(\mathtt{x})$ has a path from $\mathtt{x}_0$ to $\mathtt{x}_{n+1}$, and disconnected otherwise. When robots choose their positions randomly on $\mathscr{B}$, this graph becomes a Random Geometric Graph (RGG) as stated in Section 3.7.1.

In Chapter 7, we will extend our analysis from point robots to $R$-sized ones on the boundary. Define the *position* of a robot as that of its left end, so that robot $\mathscr{R}_i$ located at $x_i$ occupies the interval $[x_i, x_i + R]$. The support of an attached robot's position is $\mathscr{B}' = [0, s - R]$ (or a subset of it), so that the robot does not extend beyond the boundary endpoint $x = s$. We define the position vector $\mathtt{x}$ of $n$ robots to be *collision-free* (CF) if there are no inter-robot collisions, i.e. each slack $s_i$ is at least $R$.

When a robot attaches to the boundary, it effectively selects the left endpoint of an *interval* of the boundary of length $R$ lying completely within the boundary. We will generally be able to abstract intervals into points, and consequently think of a SCS as the choice of multiple random *points* on the boundary. Formally, a SCS is a *one-dimensional Point Process (PP)* (Haenggi, 2012) realized on the boundary. A special case of a PP involves robots attaching to a boundary at predefined locations. We will be interested chiefly in the *Poisson Point Process (PPP)* in which robots attach independently to the boundary. To simplify our analysis, we will first work with point robots in Section 7.2 which have $R = 0$, and consequently preclude inter-robot collisions. Point robots are an idealization of *finite* robots which have nonzero diameter; they also provide useful approximations to the behavior of finite robots when $R \ll s$.

**Properties of $\mathscr{G}$**

We define a subinterval of $\mathscr{B}$ to be *covered* by a robot team iff every point in this subinterval is within the sensing range $d$ of at least one robot in the team.Define the *covered length* of $\mathscr{G}$ as the total covered length of $\mathscr{B}$. We will compute two sets of properties of $\mathscr{G}$:

1. *connectivity properties* of each SCS that we address, which include the probability of connectivity, the mean degree of a vertex in $\mathscr{G}$, the number of connected components of $\mathscr{G}$, and the covered length of $\mathscr{G}$.

2. *spatial properties* of $\mathscr{G}$, which include the pdfs of robot positions and slacks, for both randomly generated $\mathscr{G}$ and randomly generated *connected* $\mathscr{G}$. We will subsequently extend this analysis to finite robots of diameter $R$.

The connectivity properties are drawn from the literature on RGGs (Penrose, 2003). Their computation is included in the problem statement below primarily because there are exact formulas for them in the form of integrals over polytopes, as we shall see in Chapter 7. However, the presence of these formulas is more of theoretical than practical interest, for they are laborious to compute even for the simplest cases. Nonetheless, it is remarkable that these formulas exist for attachment to a one-dimensional boundary, and can be defined so precisely. In higher dimensions such an exact approach fails, and these properties can be computed only in an asymptotic sense.

### *6.1.2 Problem Statement*

We require the robots to perform the following tasks, when placed at random in a bounded environment in which the boundary $\mathscr{B}$ is placed arbitrarily.

1. Form a connected network at the white endpoint of the boundary $\mathscr{B}$.

2. Attach to the boundary, forming a connected network, *or* cover as much of the boundary as possible.

3. Efficiently compute the list of positions taken up by the team on $\mathscr{B}$.

4. Update the map efficiently as robots join and leave $\mathscr{B}$.

Further, to analyze the system, we need to:

5. Compute the connectivity and spatial properties of $\mathscr{G}$ described in Section 6.1.1 for a random attachment scenario, focusing particularly on $p_{\mathsf{con}}(\mathscr{G})$.

6. Compute the spatial properties for a randomly chosen *connected* instance of $\mathscr{G}$.

### 6.1.3 Summary of Results

Section 6.2 will answer Parts 1-5 of the problem statement in Section 6.1.2. The results therein are readily adapted to any SCS of interest. The rest of this chapter will address uniform BC by point robots, thereby answering Parts 1-6 of the problem stated in Section 6.1.2, with a focus on point robots attaching uniformly randomly to $\mathscr{B}$. We will compute $p_{\mathsf{con}}(\mathscr{G})$ for uniform coverage by subdividing a polytope into simplices and combining their volumes in Section 6.4. Subsequently, we will compute the spatial properties of a random $\mathscr{G}$ in Section 6.5 using results from order statistics. We will then apply the decomposition of Section 6.4 to the pdfs obtained in Section 6.5 to formulate an algorithm to compute the spatial properties of a uniformly randomly sampled connected $\mathscr{G}$ in Section 6.6 and Section 6.7. These formulas are exhaustively validated in Section 6.8 with Monte Carlo simulations.

## 6.2 Deterministic Coverage Strategy (DCS) for $\mathscr{B}$

We will first provide a DCS for a group of finite-sized robots to form a connected network with uniform inter-robot spacing along a boundary $\mathscr{B}$. We adopt this approach because a DCS illustrates the working of a coverage scheme, while being simple enough to avoid the details required to define an SCS. This algorithm starts with a simple procedure, detailed in Algorithm 1 below, that is guaranteed to make all robots join the same network. Assuming

there are no faults, Algorithm 1 will terminate with all robots joining the network created by robot $\mathscr{R}_1$.

**Algorithm 1**

```
1:  procedure FORM A CONNECTED NETWORK(i)                    ▷ i = robot id
2:      state ← EXPLORE
3:      while Boundary not seen do
4:          Execute Lawnmower walk
5:      end while
6:      while White endpoint not seen do
7:          Traverse 𝒷
8:      end while
9:      if i thend i = 1
10:         Create Wi-fi network
11:     else
12:         while network not detected do
13:             Wait
14:         end while
15:         Join Wi-fi network
16:     end if
17:     state ← CONNECTED
18: end procedure
```

In this MANET, *every* node acts as a router. Once the robot team forms a connected network after the execution of Algorithm 1, the ID of every robot in the network is determined by flooding. This set of IDs is stored in the routing table of every robot. Subsequently, one robot (say, $\mathscr{R}_1$) leaves the network to determine the length $s$ of the boundary using

its odometry, and then rejoins the network by following the boundary back to its white endpoint. The maximum number of robots that can possibly attach to $\mathscr{B}$ is

$$n_{\max} = \lfloor \frac{s-R}{R} \rfloor. \tag{6.1}$$

The minimum number of robots required to ensure connectivity is

$$n_{\min} = \lfloor \frac{s}{d} \rfloor. \tag{6.2}$$

Based on these limits, the connectivity of the robot network falls into three categories:

*Case 1*: If $n < n_{\min}$, then at most $nd$ of the boundary can be covered.

*Case 2*: If $n \in [n_{\min}, n_{\max}]$, then all robots can be accommodated, and can cover the boundary entirely.

*Case 3*: If $n > n_{\max}$, then $n - n_{\max}$ robots have to be dropped from coverage. In this case, the first $n_{\max}$ robots attach to the boundary, and the remaining are dropped.

Robots subsequently take up positions that are spaced $d$ apart, so that $\mathrm{x}_i = (i-1)d$, using their odometry, with the white endpoint being considered $\mathrm{x}_0 = 0$. Afterwards, the robots can coordinate to attach to, or detach from, $\mathscr{B}$. This DCS can be easily converted to any SCS as follows. Instead of taking up equidistant positions, the robot team collectively samples from a joint pdf of their positions on $\mathscr{B}$, and attaches to these positions in order. The initial step of forming a connected network makes it possible to execute either coverage scheme by enabling the robots to share information about the boundary length and coordinate their positions along $\mathscr{B}$.

### 6.2.1 Deciding Connectivity of $\mathscr{G}$

The robots can use the *Optimized Link State Routing (OLSR)* protocol (Internet Engineering Task Force, 2014; Jacquet *et al.*, 2001) to determine the connectivity, coverage length, and number of edges of $\mathscr{G}$ at any instant. OLSR is a proactive, table-driven routing protocol, each of whose nodes maintains a table of 1-hop neighbors, which are found by flooding `HELLO` messages through the network. When a new node joins or an existing one leaves the network, a set of `TC` (Topology Control) messages are initiated by the neighbors of this node, flooding the network with updated routing tables. Robots can determine the connectivity of the network as follows: Every robot floods the network with a message consisting of its id and its position. The flooding of the network is deemed to stop after a timeout $\tau$, known to all robots, at which time every robot compiles a table of robot positions and id's. From this table, the leftmost and rightmost robot id's, located at $\mathtt{x}_1$ and $\mathtt{x}_n$, are identified. If $\mathtt{x}_1 \leq d$ and $\mathtt{x}_n \geq s - d$, then the entire network is connected. Otherwise, each robot deems the network to be disconnected as a whole.

### 6.2.2 Maintaining a List of Robot Positions

Each robot $\mathscr{R}_i$ in the network maintains data about robot positions along the boundary in the form of an *interval tree* (Berg *et al.*, 2008). If this position data is too large to fit into the memory of a robot, it will keep track only of its $m$ nearest neighbors, where the size $m$ is the maximum allowable size of the tree. The interval tree handles insertions, deletions and search queries in $O(\log m)$ time.

An incoming robot that wishes to attach to $\mathscr{B}$, say $\mathscr{R}_i$, will approach $\mathscr{B}$ and send a broadcast query to the network to determine the locations of slacks that are large enough for it to attach. A subset of the attached robots will then respond to $\mathscr{R}_i$ with a list of slacks where it may attach. Subsequently, $\mathscr{R}_i$ attaches and broadcasts its position to its neighbors, who in turn update their position data. Likewise, an outgoing robot $\mathscr{R}_i$ notifies its neighbors of its impending detachment. The neighbors recompute the resulting slacks, making note of any disconnected slacks introduced by the detachment of $\mathscr{R}_i$. They subsequently clear $\mathscr{R}_i$ for detachment, following which $\mathscr{R}_i$ detaches.

## 6.3 Slack and Position Simplices

We will now transition to considering SCS of point robots. The geometric concepts developed in this section are applicable to all SCS, regardless of the joint pdf they are defined by. Let $x_i$ be the position chosen by robot $\mathscr{R}_i$. We define the *unordered position vector* as $\mathbf{x}_{1:n}$. By sorting this vector in ascending order, we obtain the *position vector* $\mathtt{x}_{1:n}$. Figure 6.1 shows the resulting robot configuration, along with two artificial positions $\mathtt{x}_0 = 0$ and $\mathtt{x}_{n+1} = 1$. We define the $i$-th *slack* as the distance from $\mathtt{x}_{i-1}$ to $\mathtt{x}_i$:

$$s_i = \mathtt{x}_i - \mathtt{x}_{i-1}, \quad i = 1, \ldots, n+1. \tag{6.3}$$

Note that every slack is nonnegative. The *slack vector* is defined as:

$$\mathbf{s}_{1:n+1} = \mathtt{x}_{1:n+1} - \mathtt{x}_{0:n}, \tag{6.4}$$

which can be written as

$$\mathbf{s} = \mathbb{T}\,\mathtt{x}, \tag{6.5}$$

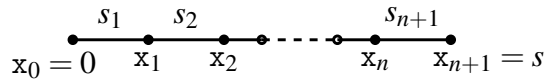



**Fig. 6.1:** Configuration of $n$ robots on $\mathscr{B} = [0, s]$, with artificial positions $\mathtt{x}_0 = 0, \mathtt{x}_{n+1} = s$ inserted.

where $\mathbb{T}$ is an $(n+1) \times (n+1)$ matrix with entries:

$$\mathbb{T}_{ij} = \begin{cases} -1, \text{ for } i = j, \\ 1, \text{ for } i = j+1, \\ 0, \text{ otherwise.} \end{cases} \tag{6.6}$$

To derive $p_{\mathsf{con}}$, we describe the geometrical objects that comprise the sets of valid position vectors $\mathtt{x}$ and slack vectors $\mathbf{s}$. Valid robot positions must satisfy the constraints $0 \leq \mathtt{x}_i \leq \mathtt{x}_{i+1} \leq s$, which can be written in vector form as:

$$\begin{aligned} \mathbf{0} \leq \mathtt{x} \leq s\mathbf{1}, \\ \mathtt{x}_{0:n} \leq \mathtt{x}_{1:n+1}. \end{aligned} \tag{6.7}$$

The set of valid position vectors forms a simplex embedded in $\mathbb{R}^n$ called the *position simplex* $\mathscr{P}$, which has vertex matrix

$$\begin{aligned} \mathbb{V}(\mathscr{P}) &= \begin{bmatrix} x_{v1} & \dots & x_{vi} & \dots & x_{v(n+1)} \end{bmatrix} \\ &= \begin{bmatrix} 0 & s\hat{e}_1 & s(\hat{e}_1 + \hat{e}_2) & \dots s(\hat{e}_1 + \dots + \hat{e}_n) \end{bmatrix}. \end{aligned} \tag{6.8}$$

where $\hat{e}_i$ denotes a unit vector along the $i$-th axis. A slack vector $\mathbf{s}$ represents a valid configuration if it satisfies the constraints:

$$\mathbf{s} \geq \mathbf{0} \text{ and } \mathbf{1}^T \mathbf{s} = s, \tag{6.9}$$

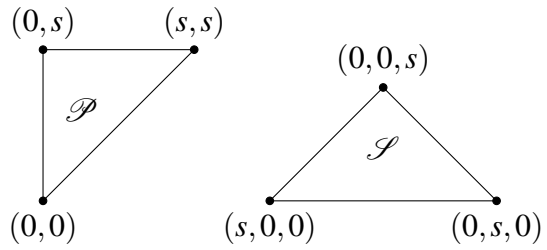



**Fig. 6.2:** Position and Slack Simplices for $n = 2$ Robots.

which can be written in full dimensional form as

$$\mathbf{s} \geq \mathbf{0} \text{ and } \mathbf{1}^T \mathbf{s}_{1:n} \leq s. \tag{6.10}$$

The set of valid slack vectors defines a simplex embedded in $\mathbb{R}^n$ called the *slack simplex* $\mathscr{S}$, which has vertex matrix:

$$\mathbb{V}(\mathscr{S}) = \begin{bmatrix} v_1 & v_2 & \dots & v_{(n+1)} \end{bmatrix} = s\mathbb{I}_{n+1}, \tag{6.11}$$

where $\mathbb{I}_n$ is the $n \times n$ identity matrix. Figure 6.2 illustrates $\mathscr{P}$ and $\mathscr{S}$ for $n = 2$ robots.

## 6.4 Computing the Probability of Connectivity

In this section, we address parts 6 and 7 of the problem statement in Section 6.1.2 for the special case of $n$ point robots on a boundary of length $s$, each attaching uniformly randomly to $\mathscr{B}$. We will now summarize the derivation of $p_{\mathsf{con}}$ that we developed in (Kumar and Berman, 2014). We then formulate an algorithm in Section 6.4.3 for estimating $p_{\mathsf{con}}$ using a Monte Carlo rejection sampling method. We apply this algorithm in Section 6.8 to check our derivation and to validate more efficient algorithms for sampling connected configurations. To compute $p_{\mathsf{con}}$, we could attempt to find the volume of the connected subset of $\mathscr{P}$ and divide it by the $n$-dimensional volume of $\mathscr{P}$, which is given by the Cayley-

Menger determinant:

$$\mathrm{Vol}(\mathscr{P}) = \frac{s^n}{n!}. \tag{6.12}$$

However, computing the volume of the connected subset involves a multi-fold integral that is complicated to evaluate in general (Kumar and Berman, 2014). Instead, we adopt a combinatorial approach that uses the slack vectors. Unlike the positions $\mathrm{x}_i$, the slacks $s_i$ are not distinguished from one another by an ordering. Consequently, $\mathscr{S}$ is a regular simplex with side length $s\sqrt{2}$, and volume

$$\mathrm{Vol}(\mathscr{S}) = \frac{\sqrt{n+1}s^n}{n!}. \tag{6.13}$$

Next, we define the space that consists only of connected slack vectors, including invalid ones that do not satisfy Equation (6.9). Geometrically, this space is an $(n+1)$-dimensional hypercube with side length $d$:

$$\mathscr{H} = \{\mathbf{s} : \mathbf{0} \leq \mathbf{s} \leq d\mathbf{1}\}. \tag{6.14}$$

All slack vectors that are both valid and connected lie in the *connected region* $\mathscr{F} := \mathscr{S} \cap \mathscr{H}$. We denote the event that $\mathbf{s} \in \mathscr{F}$ by $\mathsf{Con}$. Define $\mathscr{U} := \mathscr{S} \setminus \mathscr{F}$ to be the disconnected exterior of $\mathscr{F}$. We distinguish three ranges for $d$, of which only the second will be considered from now on:

$$d \in \begin{cases} [0, \frac{s}{n+1}], \text{ for which } \mathscr{F} = \varnothing,\ p_{\mathsf{con}} = 0; \\ (\frac{s}{n+1}, 1), \text{ for which } p_{\mathsf{con}} \in (0,1); \\ [s, \infty), \text{ for which } \mathscr{F} = \mathscr{S},\ p_{\mathsf{con}} = 1. \end{cases} \tag{6.15}$$

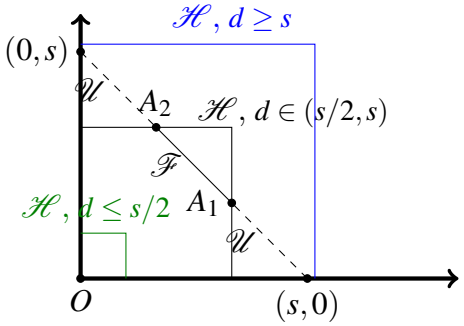



**Fig. 6.3:** Plots of $\mathscr{S} = (0,s)$–$(s,0)$ and $\mathscr{H}$ for $n = 1$. For each of the three distinct ranges of $d$ in Equation (6.15), $\mathscr{H}$ has a different type of intersection with $\mathscr{S}$. Only when $d \in (s/2,s)$ is this intersection nontrivial.

**Example 6.4.1.** *Figure 6.3 shows the case $n = 1$, for which $\mathscr{S}$ is the line segment $(0,s)$–$(s,0)$ and $\mathscr{H}$ is a square of side length $d$. When $d \in (s/2,s)$, $\mathscr{H}$ intersects $\mathscr{S}$ at two distinct points, $A_1$ and $A_2$. Thus, $\mathscr{F} = A_1$–$A_2$ and $\mathscr{U}$ is the union of the two dotted-line segments, $(0,s)$–$A_2 \cup (s,0)$–$A_1$.*

We will describe the shape of $\mathscr{F}$ in Section 6.6. Here, we compute $\mathrm{Vol}(\mathscr{U})$ indirectly using a combinatorial argument, whose details are given in Section 6.4.1. This approach relies on the decomposition of $\mathscr{U}$ into a set of overlapping regular simplices. The volumes of these simplices can be combined using the *Inclusion-Exclusion Principle* (van Lint and Wilson, 2001, Chap. 11) to yield

$$p_{\mathsf{con}} = 1 - \frac{\mathrm{Vol}(\mathscr{U})}{\mathrm{Vol}(\mathscr{S})} \tag{6.16}$$

$$= = 1 - \sum_{k=1}^{n_{\min}} (-1)^{k-1} \binom{n+1}{k} \frac{(1-kd)^n \sqrt{n+1}}{n!}. \tag{6.17}$$

Here, $n_{\min} := \lfloor s/d \rfloor$ is the minimum number of robots needed to ensure connectivity of $\mathscr{B}$, as well as the maximum possible number of disconnected slacks.

### 6.4.1 Decomposing $\mathscr{U}$ into Simplices

A naive algorithm that triangulates $\mathscr{F}$ into simplices will take a long time in practice when $n$ is large. Instead, we may write $\text{Vol}(\mathscr{F}) = \text{Vol}(\mathscr{S}) - \text{Vol}(\mathscr{U})$, decompose $\mathscr{U}$ into *overlapping* simplices, whose measures we will combine using the inclusion-exclusion principle (IEP). Suppose that $\mathbf{v}$ is a nonzero $n$-bit vector, i.e. one that has at least a single bit set. Call a slack vector $\mathbf{s} \in \mathscr{S}$ *compatible* with $\mathbf{v}$ iff $s_i$ is disconnected whenever $v_i = 1$, or $\mathbf{s}^T\mathbf{v} \geq \mathbf{1}^T\mathbf{v}$. Compatible slack vectors for $\mathbf{v}$ pose no constraints on those slacks $s_j$ for which $v_j = 0$, which could be either connected or disconnected. Let $\mathscr{U}(\mathbf{v})$ be the set of slacks that are compatible with $\mathbf{v}$. We show in (Kumar and Berman, 2014) that $\mathscr{U}$ can be written as

$$\mathscr{U} = \bigcup_{\mathbf{v} \in \{0,1\}^n \setminus \{\mathbf{0}\}} \mathscr{U}(\mathbf{v}). \tag{6.18}$$

Further, $\mathscr{U}(\mathbf{v})$ is shown to be a regular simplex having the hyperplane form (see Section 3.3)

$$\mathscr{U}(\mathbf{v}) := \{\mathbf{s} \in \mathscr{S} : \mathbf{1}^T\mathbf{s} = s - kd\}, \text{ where } k := \mathbf{1}^T\mathbf{v}. \tag{6.19}$$

Thus, $\mathscr{U}(\mathbf{v})$ has the same form as $\mathscr{S}$ itself, except that it has slack vectors summing to a shorter total slack, and has the volume

$$\text{Vol}(\mathscr{U}(\mathbf{v})) := \frac{\sqrt{n+1}(s-kd)^n}{n!}. \tag{6.20}$$

Further, $\mathscr{U}(\mathbf{v})$ is easily seen to have the vertex form:

$$\mathbb{V}(\mathscr{U}(\mathbf{v})) = (s - d\mathbf{1}^T\mathbf{v})\mathbb{I}_{n+1} + \mathbf{v}. \tag{6.21}$$

Note that only those $\mathbf{v}$ vectors with at most $n_{\min}$ 1's in them need be considered, for otherwise their compatible simplices would lie outside $\mathscr{S}$. Compatible simplices overlap, so the sum of their measures exceeds that of $\mathscr{U}$. We first decompose $\mathscr{U}$ using the inclusion-exclusion principle (IEP) as:

$$\mathrm{Vol}(\mathscr{U}) = \sum_{\mathbf{v}\in\{0,1\}^{n+1}:1\leq\mathbf{1}^T\mathbf{v}\leq n_{\min}} (-1)^{\mathbf{1}^T\mathbf{v}}\mathrm{Vol}(\mathscr{U}(\mathbf{v})). \tag{6.22}$$

Substituting Equation (6.20) into Equation (6.22) leads us to Equation (6.16). We will use this general approach for computing *integrals* over $\mathscr{F}$ later on.

### 6.4.2 Alternative Approach to Computing $\mathrm{Vol}(\mathscr{F})$

Our method of computing $\mathrm{Vol}(\mathscr{F})$ resembles Equation (3.19), but is subtly different from it. Theorem 3.6.1 places no constraints on the last slack, but our computation for $p_{\mathsf{con}}$ requires $s_{n+1} \leq d$. Following Theorem 3.6.1, we may define $\mathscr{F}$ as the intersection of $\mathscr{S}$ with *two distinct hyperplanes*:

$$\mathscr{F} := \mathscr{H} \cap \mathscr{T}_1 \cap \mathscr{T}_2, \text{ where} \tag{6.23}$$

$$\mathscr{T}_1 := \{\mathbf{s} \in \mathbb{R}^n : \mathbf{1}^T\mathbf{s}_{1:n} \leq s\}, \tag{6.24}$$

$$\mathscr{T}_2 := \{\mathbf{s} \in \mathbb{R}^n : s - \mathbf{1}^T\mathbf{s}_{1:n} \leq d\}. \tag{6.25}$$

Our approach in Section 6.4.1 sidesteps the need for two hyperplanes, but otherwise the two methods yield the same value of $p_{\mathsf{con}}$.

### 6.4.3 Monte Carlo Method for Computing $p_{\mathsf{con}}$

**Algorithm 2** Compute $p_{\mathsf{con}}$ by Rejection Sampling

```
1:  procedure REJSAMP(d, n, trials)
2:      satcount ← 0
3:      for i := 1 ... trials do
4:          vector s[1 ... n+1] ← SAMPSIMP(n)
5:          if (max(s) ≤ d) then
6:              satcount ← satcount + 1
7:          end if
8:      end for
9:      p_con ← satcount/trials
10:     return p_con
11: end procedure
```

**Algorithm 3** Sample $\mathscr{S}$ uniformly

```
1: procedure SAMPSIMP(n)
2:     vector x[1 ... n] ← rand(0, 1)[1 ... n]      ▷ Unordered position vector
3:     vector x[1 ... n] ← sort(x)                  ▷ Sorted position vector
4:     vector s[1 ... n+1] ← [x_{1:n}, 1] − [0, x_{1:n}]   ▷ Slack vector
5:     return s
6: end procedure
```

We formulate Algorithm 2 to estimate $p_{\mathsf{con}}$ by generating a set of random robot configurations and rejecting unconnected ones. The inputs to procedure REJSAMP are the values of $d$, $n$, and `trials`, the number of Monte Carlo trials that the algorithm runs in the loop at line 3. In each trial, REJSAMP calls SAMPSIMP (Algorithm 3) to sample $\mathscr{S}$ uniformly. If the slack vector $\mathbf{s}$ returned by SAMPSIMP is saturated, a counter is incremented. Finally,

REJSAMP estimates $p_{\mathsf{con}}$ as the ratio of the number of saturated vectors to the total number of trials. We note that we may use exponential random variables to speed up Algorithm 3 so that it takes $O(n)$ time to sample $\mathscr{S}$ (David and Nagaraja, 2003, Chap.2) instead of the $O(n \log n)$ time required to sort $\mathbf{x}$.

## 6.5 Computing Densities of Positions and Slacks

We will now determine the pdfs of robot positions in a randomly chosen $\mathtt{x} \in \mathscr{P}$ and slacks in a randomly chosen $\mathbf{s} \in \mathscr{S}$. The random $\mathtt{x}$ and $\mathbf{s}$ are generated in lines 3 and 4, respectively, of Algorithm 3. We note that the position and slack vectors considered here are unconditional on connectivity; we will show in Section 6.7 that their densities are needed to compute the densities of vectors $\mathtt{x}$ and $\mathbf{s}$ that are conditional on connectivity.

Let the unordered robot position $x_i$ be a realization of the random variable $X_i$, and let $\mathbf{X}$ be the random vector of all unordered positions. We define the joint and marginal pdfs of $\mathbf{X}$ by $f_{\mathbf{X}}(\mathbf{x})$ and $f_{X_i}(x_i)$, respectively. Likewise, the vectors $\mathtt{x}$ and $\mathbf{s}$ are realizations of the random vectors $\mathtt{X}$ and $\mathbf{S}$, which have the joint pdfs $f_{\mathtt{X}}(\mathtt{x})$ and $f_{\mathbf{S}}(\mathbf{s})$ and the marginal pdfs $f_{\mathtt{X}_i}(\mathtt{x}_i)$ and $f_{S_i}(s_i)$.

Our objective is to determine $f_{\mathtt{X}_i}(\mathtt{x}_i)$ and $f_{S_i}(s_i)$ given $f_{\mathbf{X}}(\mathbf{x})$. Since each robot attaches to the boundary uniformly randomly and independently of the others, each unordered position $x_i$ is distributed according to $X_i \sim \mathrm{U}(\mathscr{B})$. However, computing the pdf of the ordered positions $\mathtt{x}_i$ requires more effort. In (Kumar and Berman, 2014), we derived a marginalization procedure to determine $f_{\mathtt{X}_i}(\mathtt{x}_i)$; here, we use an alternate approach that is extensible

to more general scenarios in which the unordered robot positions may not be uniform over $\mathscr{B}$.

### 6.5.1 Marginal PDF of $\mathtt{X}_i$

We will use terminology from order statistics, as defined in Section 3.4. Substituting $f(t) = f_{X_i}(x_i) = 1/s\mathbf{I}_{\mathscr{B}}$ and $F(t) = F_{X_i}(x_i) = x'/s\mathbf{I}_{\mathscr{B}}$ into Equation (3.13), we have that

$$f_{\mathtt{X}}(\mathtt{x}) = n!\mathbf{I}_{\mathscr{P}}, \text{ or } \mathtt{X} \sim U(\mathscr{P}) \tag{6.26}$$

where $\mathbf{I}_{(.)}$ is the indicator function that evaluates to unity in the region defined by its subscript, and zero elsewhere. Further, we get

$$f_{\mathtt{X}_i}(\mathtt{x}_i) = s \text{ Beta}(\mathtt{x}_i|i, n-i+1). \tag{6.27}$$

The joint pdf of $\mathtt{X}$ is uniform over the position simplex, since the sorted positions $\mathtt{x}_i$ are just a rearrangement of the unsorted positions $x_i$. The $i$-th order statistic has a Beta density with parameters that always sum to $n+1$. From (David and Nagaraja, 2003, Ch.3), we obtain $\mathbb{E}(\mathtt{X}_i) = s \cdot \frac{i}{n+1}$ and $mode(\mathtt{X}_i) = s \cdot \frac{i-1}{n-1}$. In addition, the covariance matrix of $\mathtt{X}$ is given by

$$\mathbb{C}(\mathtt{X})_{i,j} = \frac{i(n+1-j)}{(n+1)^2(n+2)}, \text{ for } 1 \leq i \leq j \leq n. \tag{6.28}$$

Since the entries in the lower triangular sub-matrix of $\mathbb{C}(\mathtt{X})$ with indices of the form $(i, j) : i > j$ are undefined, we make $\mathbb{C}(\mathtt{X})$ symmetric by setting $\mathbb{C}(\mathtt{X})_{i,j} := \mathbb{C}(\mathtt{X})_{j,i}$ for this sub-matrix.

That the pdf of $\mathtt{X}_i$ is a rescaled Beta density makes intuitive sense for the following reasons. First, the $i$-th largest order statistic $\mathtt{X}_{n-i+1}$ is the *reflection* of the $i$-th smallest statistic $\mathtt{X}_i$,

in that $f_{\mathtt{X}_{n-i+1}}(\mathtt{x}_{n-i+1}) = f_{\mathtt{X}_i}(1-\mathtt{x}_i)$. We may think of $\mathtt{X}_{n-i+1}$ as the $i$-th order statistic that we encounter when we move from right to left on $\mathscr{B}$; in other words, it would have been the $i$-th smallest order statistic had we chosen $1-\mathtt{x}_i$ as the argument of the pdf $f_{\mathtt{X}_i}(\mathtt{x}_i)$. We expect these two order statistics to be equal distances from the endpoint of $\mathscr{B}$ that is closest to them, which is borne out by the mirror symmetry of their Beta pdfs. Second, the mean positions $\mathbb{E}(\mathtt{X}_i) = \frac{is}{n+1}$ divide $\mathscr{B}$ into $n+1$ equal parts, as we would expect of the robot configurations on average. Finally, the modes of the pdfs $f_{\mathtt{X}_i}(\mathtt{x}_i)$ peak successively to the right, which makes sense since $\mathtt{x}_i \leq \mathtt{x}_{i+1}$.

### 6.5.2 Marginal PDF of $S_i$

We start by computing the joint density $f_{\mathbf{S}}(\mathbf{s})$ of the slack vector over $\mathscr{S}$. Since every ordered position vector is equally likely, the corresponding slack vectors are also equally likely; in other words, the random slack vector $\mathbf{S}$ is uniform over $\mathscr{S}$:

$$\mathbf{S} \sim \mathrm{U}(\mathscr{S}) \text{ or } f_{\mathbf{S}}(\mathbf{s}) = \frac{1}{\mathrm{Vol}(\mathscr{S})}\mathbf{I}_{\mathscr{S}}. \tag{6.29}$$

As noted in Section 6.4, the slacks are not subject to ordering constraints, unlike the ordered positions $\mathtt{x}_i$, which are non-decreasing. If Perm is any permutation on the entries of $\mathbf{s}$, then the joint pdf $f_{\mathrm{Perm}(\mathbf{S})}\mathrm{Perm}(\mathbf{s})$ is identical to $f_{\mathbf{S}}(\mathbf{s})$. We cannot impose a permutation on the position vector $\mathtt{x}$, since any permutation except the identity permutation will cause the sequence of ordered positions to violate the constraints $\mathtt{x}_i \leq \mathtt{x}_{i+1}$. However, since we face no such restriction with the slacks, the $s_i$ are *exchangeable* random variables. Consequently, they share the same marginal pdf $f_{S_i}(s_i)$, and so we only need to determine the marginal

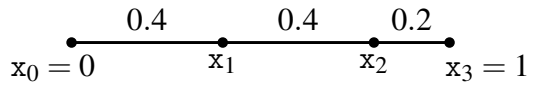



**Fig. 6.4:** The problem instance $(s = 1, n = 2, d = 0.4)$ with parameters $n_{\min} = n$, $l = 0.2$, and $z = 0$ is an instance of the *small n case*. The figure above shows a possible connected configuration, in which $s_1 = s_2 = d$ and $s_3 = l$.

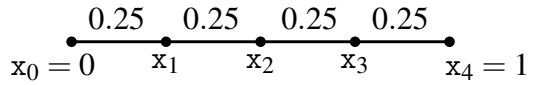



**Fig. 6.5:** The problem with $(s = 1, n = 3, d = 0.4)$ with the parameters $n_{\min} = 2$, $l = 0.2$, and $z = 1$ is an instance of the *large n case*. The figure above shows a possible connected configuration, in which each slack equals 0.25.

pdf of one of the slacks, say $s_1$. Observe that since $\mathtt{x}_1 = s_1$, we have that

$$f_{S_i}(s_i) = f_{S_1}(s_1) = f_{\mathtt{X}_1}(\mathtt{x}_1) = s\ \text{Beta}(\mathtt{x}_1|1,n), \tag{6.30}$$

and so $S_i \sim s\ \cdot \text{Beta}(s_i|1,n)$. It is easy to see that $\mathbb{E}(S_i) = \frac{s}{n+1}$; further, from Equation (6.6) we can calculate the covariance matrix $\mathbb{C}(\mathbf{S})$ as

$$\mathbb{C}(\mathbf{S}) = \mathbb{T}\ \mathbb{C}(\mathtt{X})\ \mathbb{T}^T. \tag{6.31}$$

## 6.6 Shape of $\mathscr{F}$

As shown in the previous sections, characterizing the shape of $\mathscr{S}$ enabled us to determine the pdf of $S_i$. In this section and the next, we will characterize the shape of the connected subset $\mathscr{F}$ and continue to exploit connections between geometry and probability in order to compute the marginal pdfs of $S_i$ and $\mathtt{X}_i$ over $\mathscr{F}$, as specified in part (7) of the Problem Statement, which required us to sample from connected instances of $\mathscr{G}$. We will use the notation $S_i|\mathsf{Con}$ to signify $S_i$ conditioned on connectivity, or alternatively, $S_i$ sampled from $\mathscr{F}$.

In our analysis, we consider two different cases for the robot population $n$, shown respectively in Figure 6.4 and Figure 6.5: (1) the *small n case*, in which there are just enough robots to achieve a connected configuration on the boundary, and (2) the *large n case*, in which there are more than enough robots to achieve connectivity. We now provide technical definitions of these two cases in terms of the problem parameters. Suppose that the position of each robot $\mathscr{R}_i$, $i = 1, ..., n_{\min}$, is defined as $\mathrm{x}_i = id$. Then the robot that is closest to the boundary endpoint at $x = s$ is located at $\mathrm{x}_{n_{\min}} = n_{\min} d$. The slack between this robot and the endpoint $x = s$ is the leftover length,

$$l = s - n_{\min} d. \tag{6.32}$$

There are $z \geq 0$ remaining robots, where $z$ is defined as:

$$z = \begin{cases} n - n_{\min} + 1 & \text{if } l = 0, \\ n - n_{\min} & \text{otherwise.} \end{cases} \tag{6.33}$$

When $z = 0$, there are just enough robots to connect $\mathscr{B}$, or equivalently, $d \in [\frac{s}{n+1}, \frac{s}{n})$. The condition $z = 0$ defines the *small n case*, and the condition $z > 0$ defines the *large n case*.

### 6.6.1 Vertices of $\mathscr{F}$

Any connected slack point $\mathbf{s} \in \mathscr{F}$ can have at most $n$ of its slack coordinates equal to $d$ at the same time, in which case the remaining slack coordinate becomes equal to $l$. Indeed, all "extremal" slack points will have this form. We now state and prove this result formally:

**Theorem 6.6.1.** ***(Vertices of $\mathscr{F}$ for small n)*** *For small n, the vertex matrix of $\mathscr{F}$ is given by:*

$$\mathbb{V}(\mathscr{F}) = \begin{bmatrix} v_1 & v_2 & \dots & v_{(n+1)} \end{bmatrix}$$

$$= \begin{bmatrix} l & d & \dots & d \\ d & l & \dots & d \\ \dots & & & \\ d & \dots & l & d \\ d & d & \dots & l \end{bmatrix}$$

$$= d\,\mathbf{1}_{n+1} \cdot \mathbf{1}_{n+1}^T + (l-d)\,\mathbb{I}_{n+1}. \tag{6.34}$$

*Proof.* We can write the constraints that define the set of valid connected slack vectors that comprise $\mathscr{F}$ as the following set of inequalities:

$$\begin{aligned} s_i \leq d, \qquad & -s_i \leq 0, \quad i = 1 \dots n+1, \\ \sum_{i=1}^{n+1} s_i \leq s, \qquad & \sum_{i=1}^{n+1} -s_i \leq -s. \end{aligned} \tag{6.35}$$

Since $\mathscr{F}$ can be expressed as an intersection of half-spaces, it is a convex polytope. We know that this polytope is embedded in dimension $n+1$. From (Boyd and Vandenberghe, 2004), the extreme points, or vertices, of the polytope are those for which some $(n+1)$-element, linearly independent subset of the inequalities in Equation (6.35) are replaced by equalities. The $n+1$ equations

$$\begin{aligned} s_i = d, \quad & i = 2 \dots n+1, \\ \sum_{i=1}^{n+1} s_i = s \quad & \end{aligned} \tag{6.36}$$

form one such subset, which is solved by setting $s_1 = l$ and $s_{i \neq 1} = d$. Thus, the vector $v_1 = \begin{bmatrix} l & d & \dots & d \end{bmatrix}^T$ forms one vertex of $\mathscr{F}$. The remaining vertices $v_i$, $i = 2, \dots, n+1$, can be obtained by setting $s_i = l$, $s_{j \neq i} = d$. It follows immediately that $\mathscr{F}$ is a regular simplex with side length $\sqrt{2}(d-l)$. Figure 6.6 shows an instance of $\mathscr{F}$ and $\mathscr{S}$ when $n = 2$. □

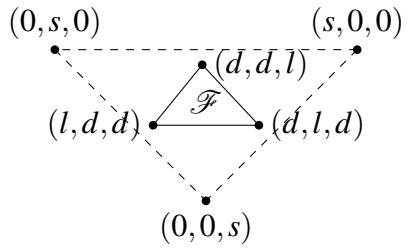



**Fig. 6.6:** Connected Region $\mathscr{F}$ in $\mathscr{S}$ for $n = 2$.

Given two vectors **a** and **b**, say that **b** is an expansion of **a** to $m$ elements iff **b** has $m$ elements, and **b** pads the entries of **a** with zeros, while retaining the order of entries of **a**. In other words, we have

$$\mathbf{b} = \begin{bmatrix} \mathbf{0} & a_1 & \mathbf{0} & a_2 & \dots & a_n & \mathbf{0} \end{bmatrix}, \tag{6.37}$$

where each **0** is a vector whose entries are all zeros, and which could possibly be empty.

**Theorem 6.6.2.** ***(Vertices of $\mathscr{F}$ for large** n)* *Every vertex of $\mathscr{F}$ is an $(n+1)$-element expansion of some vertex of $\mathscr{F}_{n_{\min}}$, i.e., it is a permutation of*

$$v_1 = \begin{bmatrix} d\mathbf{1}_K \\ l \\ \mathbf{0}_z \end{bmatrix}. \tag{6.38}$$

*The columns of the vertex matrix $\mathbb{V}(\mathscr{F})$ consist of all such permutations.*

*Proof.* We repeat the proof of Theorem 6.6.1, keeping in mind that there are now $n+1 > K+1$ slacks. Once again, the extreme points of $\mathscr{F}$ are the solutions to the set of equalities in Equation (6.36). These solutions consist of all permutations of the vector $v_1$ in Equation (6.38). □

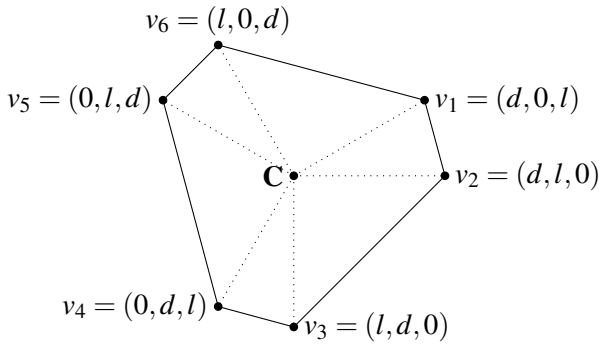



**Fig. 6.7:** Hexagonal $\mathscr{F}$ for $(s = 1, n = 2, d = 0.6)$ subdivided into triangles.

The number of vertices of $\mathscr{F}$ is given by the multinomial coefficient

$$\binom{n}{n_{\min}, 1, z} = z\binom{n}{n_{\min}}. \tag{6.39}$$

In contrast to the small $n$ case, in which $\mathscr{F}$ is a regular simplex, we find that $\mathscr{F}$ in general admits no simple geometric description. However, the vertices of $\mathscr{F}$ are equidistant from the *centroid* of $\mathscr{F}$,

$$\mathbf{C} := \frac{s\mathbf{1}}{n+1}, \tag{6.40}$$

and thus lie on an $(n-1)$-dimensional sphere centered at $\mathbf{C}$. For both cases, since the vertices of $\mathscr{F}$ are drawn from the permutations of a multiset, $\mathscr{F}$ is a *multi-permutahedron* (Billera and Sarangarajan, 1994).

**Example 6.6.1.** *Figure 6.7 shows a hexagonal $\mathscr{F}$ for the quadruple $(s = 1, n = 2, d = 0.6)$. Alternating pairs of facets (line segments) are congruent.*

## 6.7 Marginal PDFs over $\mathscr{F}$

### *6.7.1 Small n Case*

We now use a result from (Devroye, 1986) that will enable us to compute the marginal pdfs of $S_i|\mathsf{Con}$ and $\mathtt{x}_i|\mathsf{Con}$, the slacks and positions for connected robot configurations.

**Theorem 6.7.1.** ***(Sampling a simplex)*** *Let* $\mathbf{w}$ *represent the slack vector generated by uniformly sampling n points on* $\mathscr{I}$*, and let* $\mathscr{S}$ *be any* $(n+1)$*-vertex simplex embedded in* $\mathbb{R}^n$*. Then the product* $\mathbf{s} = \mathbb{V}(\mathscr{S})^T\mathbf{w}$ *generates a uniformly random slack vector on* $\mathscr{S}$*.*

The slack vector $\mathbf{w}$, whose purpose is to weight the vertex vectors, is called a *weighted slack vector*, and its elements $s_{wi}$ are called the weighted slacks. Applying Theorem 6.7.1 to the simplex $\mathscr{F}$ yields $\mathbf{s}|\mathsf{Con} = \mathbb{V}(\mathscr{F})^T\mathbf{w}$. Hence, sampling $\mathscr{F}$ will take $O(n^2)$ time, which is the time required to compute the product $\mathbb{V}(\mathscr{F})^T\mathbf{w}$. Using the definition of $\mathbb{V}(\mathscr{F})$ in Equation (6.34), we obtain

$$s_i|\mathsf{Con} = d-(d-l)s_{wi}, \quad i = 1\ldots n+1. \tag{6.41}$$

From this result, we see that each $s_i|\mathsf{Con}$ lies in $[l,d]$. The weighted slacks $s_{wi}$ have the same distributions as the slacks $s_i$ of Equation (6.30). Using this fact and Equation (6.41), we find that the pdf of $S_i|\mathsf{Con}$ is a shifted and scaled Beta pdf with its *sign changed*:

$$S_i|\mathsf{Con} \sim d-(d-l)\text{Beta}(s_i|1,n). \tag{6.42}$$

The *weighted positions*, defined as $x_{wi} := \sum_{j=1}^{i} s_{wj}$, have the same distributions Equation (6.27) as the positions $\mathtt{x}_i$. Using this fact along with Equation (6.41) and Equa-

tion (6.42), we obtain:

$$\mathtt{x}_i|\mathsf{Con} = id - i(d-l)x_{wi}, \tag{6.43}$$

$$\mathtt{X}_i|\mathsf{Con} \sim id - i(d-l)\text{Beta}(\mathtt{x}_i|i, n-i+1). \tag{6.44}$$

### 6.7.2 *Large n Case*

Unlike with small $n$, the large $n$ case does not offer any simple formulas for connected slacks and positions, for $\mathscr{F}$ is no longer a regular simplex. The consequent lack of symmetry in $\mathscr{F}$ makes it hard for formulas analogous to Equation (6.42) to exist. However, we may follow the approach of Section 6.4.1 to develop a method that computes the marginal pdfs $f_{S_1}(s_1) \sim f_{S_i}(s_i)$ over $\mathscr{F}$. This approach will aid us in Chapter 7 where we will deal with nonuniform parents. We begin by noting that

$$S_1 = s - \sum_{i=2}^{n+1} S_i \text{ over } \mathscr{S}. \tag{6.45}$$

We then have

$$f_{S_1}(s_1)|\mathsf{Con} = \int_{\mathscr{F}} (s - \sum_{i=2}^{n+1} s_i) d\mathbf{s} = s \cdot \text{Beta}(1,n) - \int_{\mathscr{U}} (s - \sum_{i=2}^{n+1} s_i) d\mathbf{s}. \tag{6.46}$$

It remains to compute the integral of $f_{S_1}(s_1)$ over $\mathscr{U}$. Let $\text{Leb}(.,.)$ denote the integral of its first argument over the region in its second argument. For any function $I$ defined on $\mathscr{S}$, we have by the same reasoning as Section 6.4.1 that

$$\text{Leb}(I, \mathscr{U}) = \sum_{\mathbf{v} \in \{0,1\}^{n+1}} (-1)^{\mathbf{1}^T \mathbf{v}} \text{Leb}(I, \mathscr{U}(\mathbf{v})). \tag{6.47}$$

Substituting $I := s - \sum_{i=2}^{n+1} s_i$ to Equation (6.47) gives us the marginal pdf of $f_{S_1}(s_1)$ on $\mathscr{U}$. Since $\mathscr{U}(\mathbf{v})$ is a simplex, any integral over it can be computed by a straightforward linear transformation from the canonical simplex in $\mathbb{R}^n$. Likewise, the order statistics $\mathtt{X}_i$ are sums of consecutive slacks, and can be found by

$$f_{\mathtt{X}_i}(\mathtt{x}_i)|\mathsf{Con} = \int_{\mathscr{F}} (s - \sum_{j=i+1}^{n+1} s_j) d\mathbf{s} = s \cdot \mathrm{Beta}(1,n) - \int_{\mathscr{U}} (s - \sum_{j=2}^{i+1} s_j) d\mathbf{s}. \tag{6.48}$$

The formulas for $f_{\mathtt{X}_i}(\mathtt{x}_i)$ and $f_{S_i}(s_i)$ at connectivity have no simple form that we are aware of. Yet, it is evident that they are mixtures of rescaled Beta pdfs, of which Equation (6.44) is an example. We will compute the pdfs of $f_{S_i}(s_i)$ for a special case in Section 6.8. Note that the expected slack vector over $\mathscr{F}$ coincides with the centroid of $\mathscr{F}$.

### 6.7.3 REPSAMP: Fast Random Sampling of $\mathscr{F}$

Given the labor of directly computing the integrals in Equation (6.48) and Equation (6.46), we now outline a way to approximate these pdfs by sampling over $\mathscr{F}$ in $O(n^2)$ time in Algorithm 4.

We offer here a heuristic explanation of why Algorithm 4 generates uniform random samples of $\mathscr{F}$; in future work, we will investigate its theoretical underpinnings. In Section 6.8, we verify through Monte Carlo simulations that the slack vectors generated by Algorithm 2 and Algorithm 4 are identical in distribution.

We start by considering an arbitrary *disconnected* slack vector, $\mathbf{s} \in \mathscr{U}$. We may deduce from Equation (6.33) that at most $n_{\min}$ of the slacks in $\mathbf{s}$ can be disconnected. Thus, the remaining $z+1$ slacks will be connected, *regardless* of the choice of $\mathbf{s}$. In other words, for

**Algorithm 4** REPSAMP: Fast random sampling of $\mathscr{F}$

**procedure** REPSAMP$(s,d,n)$
    Compute $n_{\min}, l, z, \mathbb{V}(\mathscr{F}_{n_{\min}})$ as described in text
    Compute $\bar{\mathbf{s}}_{1:K+1}$ by sampling $\mathscr{F}_{n_{\min}}$ (Theorem 6.7.1)
    Compute $\bar{\mathrm{x}}_{1:K}$ from $\bar{\mathbf{s}}$
    $\mathbf{s}^r_{1:z+1} \leftarrow$ SAMPSIMP$(z)$
    Compute $\mathrm{x}^r_{1:z}$ from $s \cdot \mathbf{s}^r$
    $\mathrm{x} \leftarrow$ merge $(\bar{\mathrm{x}},\ \mathrm{x}^r,\ \mathrm{x}_0 = 0,\ \mathrm{x}_{n+1} = s)$
    Return $\mathbf{s} \leftarrow \mathrm{x}_{1:n+1} - \mathrm{x}_{0:n}$
**end procedure**

**s** to be connected, we need to ensure that the first $n_{\min}$ slacks are connected. We will now detail how to choose the first $n_{\min}$ slacks in such a way that, immaterial of the choice of the remaining $z+1$ slacks, the configuration remains connected.

First, suppose that we need to generate a sample of a connected $n$-dimensional slack vector **s** that has $n_{\min}+1$ nonzero slacks. To do this, we first note that $\mathscr{F}_{n_{\min}}$ is a simplex. Thus, we may sample $\mathscr{F}_{n_{\min}}$ using Theorem 6.7.1 to obtain a *representative slack vector* $\bar{\mathbf{s}} \equiv \mathbf{s}_{1:K+1}$, all of whose slacks $\bar{s}_i$ are nonzero almost surely. We can then choose a random $(n+1)$-element expansion of $\bar{\mathbf{s}}$ to obtain **s**. Geometrically, this process chooses a random point on one of the facets of $\mathscr{F}$.

Now, suppose that our sample of a connected $n$-dimensional vector **s** must have $n_{\min}+2$ nonzero slacks. From the previously generated $(n_{\min}+1)$-dimensional vector $\bar{\mathbf{s}}$, we can randomly choose slack $\bar{s}_i$ with probability $p_i = \bar{s}_i/s$. We split $\bar{s}_i$ into two slacks $\bar{s}_{i,1}$ and $\bar{s}_{i,2} = \bar{s}_i - \bar{s}_{i,1}$, where $\bar{s}_{i,1}$ is uniformly random on the interval $[0,\ \bar{s}_i]$. Now we increment the

indices of slacks $\bar{s}_{i+1:n_{\min}+1}$ by one and insert slacks $\bar{s}_{i,1}$ and $\bar{s}_{i,2}$ as $\bar{s}_i$ and $\bar{s}_{i+1}$, respectively. The resulting $\bar{\mathbf{s}}$ has $n_{\min}+2$ nonzero slacks. A random $(n+1)$-element expansion of this $\bar{\mathbf{s}}$ yields $\mathbf{s}$, which is a random point on the 1-simplex between two randomly chosen facets of $\mathscr{F}$.

By induction, if we need to generate a random $\mathbf{s}$ with $n_{\min}+1+z$ nonzero slacks, we may split an arbitrary number of $\bar{s}_i$ using $z$ random points. This will give us a random point on the $z$-dimensional simplex that connects $z+1$ randomly chosen facets of $\mathscr{F}$. Thus this procedure results in uniform random sampling of the entirety of $\mathscr{F}$.

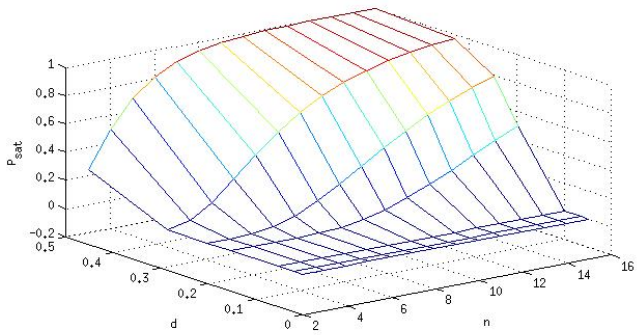
Algorithm 4 outlines the procedure described above. Line 3 samples the representative slack vector $\bar{\mathbf{s}}$ from $\mathscr{F}_{n_{\min}}$ using Theorem 6.7.1. A representative position vector $\bar{\mathtt{x}}_{1:n_{\min}}$ is then generated from the partial sums of $\bar{\mathbf{s}}$. There are now $z$ intermediate positions remaining to be chosen. In line 5, we use Algorithm 3 to generate a slack vector $\mathbf{s}^r_{1:z+1}$, which we multiply by the boundary length $s$ and use to compute the intermediate positions $\mathtt{x}^r_{1:z}$. Finally, the two vectors $\bar{\mathtt{x}}$ and $\mathtt{x}^r$ and the artificial positions $\mathtt{x}_0$ and $\mathtt{x}_{n+1}$ are merged in sorted order, and the corresponding slack vector $\mathbf{s}$ is computed and returned. The algorithm's execution time is determined by line 3, which takes $O(n^2_{\min})$ time, and line 7, which takes $O(z\log z+n)$ time. Thus, Algorithm 4 runs in $O(n^2_{\min}+z\log z+n)$ time.

## 6.8 Validation

In this section, we address part (6) of the Problem Statement by validating each of our analytical results against the output of $N=25000$ Monte Carlo simulations. These simulations were coded in MATLAB and used MATLAB's distribution fitting toolbox `dfittool`

(MathWorks , Inc., 2014) to estimate pdfs. Consequently, we consider only point robots in the subsequent sections.

**Fig. 6.8:** 3-D plot of $p_{\mathsf{con}}$ against input arguments $(d,n)$ with $s = 1$




### 6.8.1 Probability of Connectivity, $p_{\mathsf{con}}$

Table 6.2 compares $p_{\mathsf{con}}$ from Equation (6.16) with the estimated $\hat{p}_{\mathsf{con}}$ obtained from $N$ trials of Algorithm 2. The left and right subtables show $p_{\mathsf{con}}$ varying with $d$ and $n$, respectively. Each table in this section will follow the format of Table 6.2 by specifying the problem parameters in the first column and comparing analytically computed parameters against their estimated values. Further, Figure 6.8 shows how $p_{\mathsf{con}}$ varies with input arguments $d$ and $n$.

### 6.8.2 Marginal PDFs of $\mathtt{X}_i$ and $S_i$

This test verifies the formulas for the marginal pdfs of Equation (6.30) and Equation (6.27), both of which are unconditional upon connectivity. In Table 6.3, the parameters $(a,b)$ of the Beta pdfs in these equations are compared with their estimated counterparts $(\hat{a},\hat{b})$, computed from $N$ runs of Algorithm 3.

**Table 6.2:** Point Robots: $p_{\mathsf{con}}$ vs. $\hat{p}_{\mathsf{con}}$

| $s=1$ $n=16$ | | | $s=1$ $d=0.5$ | | |
|---|---|---|---|---|---|
| $d$ | $p_{\mathsf{con}}$ | $\hat{p}_{\mathsf{con}}$ | $n$ | $p_{\mathsf{con}}$ | $\hat{p}_{\mathsf{con}}$ |
| 0.05 | 0.0000 | 0.0000 | 3 | 0.2500 | 0.2467 |
| 0.10 | 0.0001 | 0.0001 | 4 | 0.5000 | 0.5029 |
| 0.20 | 0.4929 | 0.4925 | 5 | 0.6875 | 0.6885 |
| 0.25 | 0.7898 | 0.7913 | 6 | 0.8125 | 0.8124 |
| 0.33 | 0.9635 | 0.9651 | 7 | 0.8906 | 0.8899 |
| 0.50 | 0.9995 | 0.9997 | 8 | 0.9375 | 0.9383 |

**Table 6.3:** Estimating $\mathtt{X}$ and $\mathbf{S}$

| $s=1$ | | | $n=4$ | |
|---|---|---|---|---|
| Variable | $a$ | $b$ | $\hat{a}$ | $\hat{b}$ |
| $S_1=\mathtt{X}_1$ | 1 | 4 | 0.999 | 3.987 |
| $\mathtt{X}_2$ | 2 | 3 | 2.004 | 2.991 |
| $\mathtt{X}_3$ | 3 | 2 | 2.958 | 1.960 |
| $\mathtt{X}_4$ | 4 | 1 | 4.010 | 1.002 |

### *6.8.3 Covariance Matrices of Positions and Slacks*

Tables 6.4 and 6.5 compare the covariance matrices $\mathbb{C}(\mathtt{X})$ and $\mathbb{C}(\mathbf{S})$ computed from Equation (6.28) and Equation (6.31) with their estimates, which were obtained from $N$ runs of Algorithm 3.

**Table 6.4:** Estimating Position Covariance Matrices

| | $s=1$ $n=3$ | | |
|---|---|---|---|
| $\mathbb{C}(\mathtt{X}) =$ | 0.037 | 0.025 | 0.0125 |
| | 0.025 | 0.050 | 0.0250 |
| | 0.012 | 0.025 | 0.037 |
| $\hat{\mathbb{C}}(\mathtt{X}) =$ | 0.037 | 0.025 | 0.013 |
| | 0.025 | 0.050 | 0.025 |
| | 0.012 | 0.026 | 0.038 |

**Table 6.5:** Estimating Slack Covariance Matrices

| | $s=1$ $n=3$ | | | |
|---|---|---|---|---|
| $\mathbb{C}(\mathbf{S}) =$ | 0.037 | −0.012 | −0.012 | −0.012 |
| | −0.013 | 0.037 | −0.012 | −0.012 |
| | −0.013 | −0.012 | 0.037 | −0.012 |
| | −0.013 | −0.012 | −0.012 | 0.038 |
| $\hat{\mathbb{C}}(\mathbf{S}) =$ | 0.037 | −0.012 | −0.013 | −0.012 |
| | −0.012 | 0.037 | −0.013 | −0.012 |
| | −0.013 | −0.013 | 0.038 | −0.013 |
| | −0.012 | −0.012 | −0.013 | 0.037 |

### *6.8.4 Statistics at Connectivity: Small n*

We validate our formulas for the statistics of the *small n case*, described in Section 6.7. Table 6.6 compares the predicted parameter values of the Beta pdfs of $\mathbf{S}|\mathsf{Con}$ and $\mathtt{X}|\mathsf{Con}$, given by Equation (6.42) and Equation (6.44), with their estimates. These estimates were computed by applying Theorem 6.6.1 to slack vectors generated from $N$ runs of Algorithm 3.

Table 6.7 compares the covariance matrix of $\mathbf{S}|\mathsf{Con}$ with its counterpart estimated from the same Monte Carlo simulation data. Note that the estimated parameters $\hat{a}, \hat{b}$ are the interchanged versions of their true counterparts, as the rescaled Beta pdfs of Equation (6.42) and Equation (6.43) have negative signs. For further validation, we plotted the frequency of two robot positions, $\mathtt{X}_2|\mathsf{Con}$ and $\mathtt{X}_3|\mathsf{Con}$, that were generated by the Monte Carlo simulations. As Figure 6.9 shows, both frequency plots can be fit to the Beta pdf that is predicted by Equation (6.43).

**Table 6.6:** Estimating $\mathbf{S}|\mathsf{Con}$ and $\mathtt{X}|\mathsf{Con}$ for small $n$

| $s=1$ | | | $n=3$ | $d=0.3$ |
|---|---|---|---|---|
| Variable | $a$ | $b$ | $\hat{b}$ | $\hat{a}$ |
| $S_1, \mathtt{X}_1|\mathsf{Con}$ | 3 | 1 | 2.860 | 0.991 |
| $\mathtt{X}_2|\mathsf{Con}$ | 2 | 2 | 1.969 | 1.982 |
| $\mathtt{X}_3|\mathsf{Con}$ | 1 | 3 | 0.986 | 2.826 |

**Table 6.7:** Estimating Slack Covariance Matrices for Small $n$

$$\mathbb{C}(\mathbf{S}|\mathsf{Con}) = \begin{matrix} s=1 \\ \\ \\ \\ \end{matrix} \begin{matrix} n=3 \\ \\ \\ \\ \end{matrix} \begin{matrix} d=0.3 \\ \\ \\ \\ \end{matrix} \begin{bmatrix} 0.0015 & -0.0005 & -0.0005 & -0.0005 \\ -0.0005 & 0.0015 & -0.0005 & -0.0005 \\ -0.0005 & -0.0005 & 0.0015 & -0.0005 \\ -0.0005 & -0.0005 & -0.0005 & 0.0015 \end{bmatrix}$$

$$\hat{\mathbb{C}}(\mathbf{S}|\mathsf{Con}) = \begin{bmatrix} 0.0015 & -0.0005 & -0.0005 & -0.0005 \\ -0.0005 & 0.0015 & -0.0005 & -0.0005 \\ -0.0005 & -0.0005 & 0.0015 & -0.0005 \\ -0.0005 & -0.0005 & -0.0005 & 0.0015 \end{bmatrix}$$

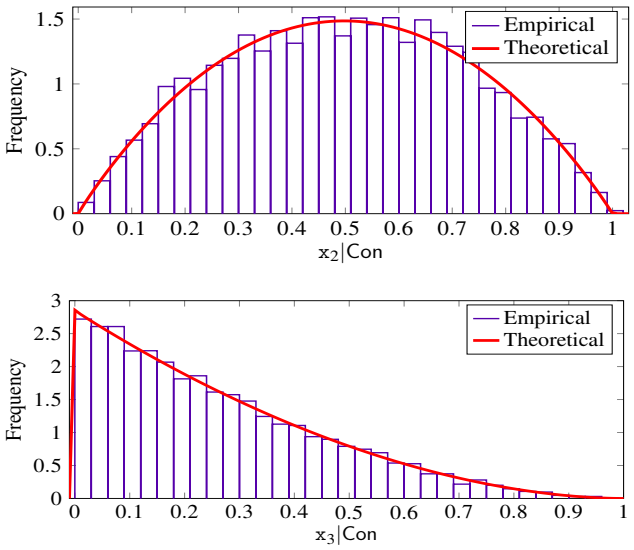



**Fig. 6.9:** Frequency plots of $N = 25000$ samples of $\mathsf{x}_2|\mathsf{Con}$ and $\mathsf{x}_3|\mathsf{Con}$ from Monte Carlo trials. Both plots are fit to the Beta pdf given by Equation (6.44).

### *6.8.5 Statistics at Connectivity: Large n*

We now validate the formulas of Section 6.7 for the problem instance $Q(s = 1, n = 2, d = 0.6)$ from Figure 6.7, which has the parameters $n_{\min} = 1$, $l = 0.4$, and $z = 1$.

**Expected value of $\mathbf{S}|\mathsf{Con}$**

We generated connected slack vectors from $N$ runs of Algorithm 2. The mean of these saturated slack vectors was

$$\hat{\mathbf{C}} = \begin{bmatrix} 0.3324 & 0.3323 & 0.3353 \end{bmatrix}^T, \tag{6.49}$$

which agrees closely with the centroid $\mathbf{C} = \frac{1}{3}\mathbf{1}$.

**Marginal pdfs of Slacks at Connectivity**

We follow (Kumar and Berman, 2016b) to subdivide $\mathscr{F}$ into triangles, and compute slacks over each triangle. We consider the set of slack vectors that comprise triangle $\mathcal{P}_1$ in Figure 6.7. Applying Theorem 6.7.1 to $\mathcal{P}_1$, we can compute a uniformly random slack vector $\mathbf{s} \in \mathcal{P}_1$ as follows:

$$\mathbf{s} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} d & d & \frac{1}{3} \\ 0 & l & \frac{1}{3} \\ l & 0 & \frac{1}{3} \end{bmatrix} \begin{bmatrix} s_{w1} \\ s_{w2} \\ s_{w3} \end{bmatrix} = \begin{bmatrix} d + (\frac{1}{3} - d)s_{w3} \\ ls_{w2} + \frac{s_{w3}}{3} \\ ls_{w1} + \frac{s_{w3}}{3} \end{bmatrix}. \tag{6.50}$$

We know that each weighted slack $s_{wi}$ has a Beta$(1,2)$ density, which is a triangular pdf. From Equation (6.42), it is evident that $S_1$ has a scaled and shifted triangular pdf. However, the pdfs of the identically distributed slacks $S_2$ and $S_3$ are not obvious. To compute $f_{S_2}(s_2)$, first observe that $s_2$ has the range $[0,l]$. From $N$ runs of Algorithm 2, $s_2$ appears to have a triangular pdf with mode $\frac{1}{3}$, as shown by the frequency plot in Figure 6.10. The figure shows that this frequency plot is closely fit by a plot with two line segments, $(0,0)$–$(1/3,2/l)$ and $(1/3,2/l)$–$(l,0)$. Note that the area under this fitted plot is unity, as required for a pdf. Hence, the fitted plot is the pdf of $S_2$, which we can write as:

$$f_{S_2}(s_2) = \begin{cases} \frac{6}{l} s_2, & s_2 \in [0, \frac{1}{3}] \\ \\ \frac{6}{l(1-3l)}(s_2 - l), & s_2 \in [\frac{1}{3}, l]. \end{cases} \tag{6.51}$$

This triangular pdf is a scaled and shifted version of Beta$(1,2)$, similar to the pdf of $s_1$.

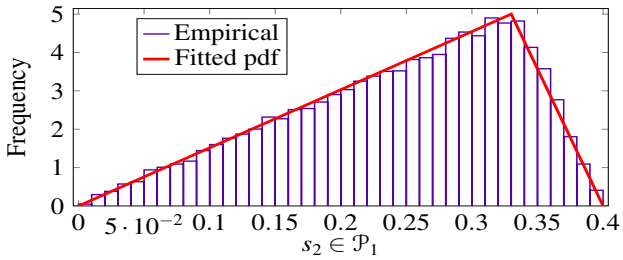



**Fig. 6.10:** Frequency plot of $N = 25000$ samples of $S_2 \in \mathcal{P}_1$ from Monte Carlo trials. The plot is fit to the Beta pdf given by Equation (6.50).

**Table 6.8:** Equivalence of Sampling Algorithms - 1

| $s = 1$ | $n = 2$ | $d = 0.6$ |
|---|---|---|
| Moment of $S_1$ | REJSAMP | REPSAMP |
| $\mathbb{E}(S_1)$ | 0.3317 | 0.3317 |
| $\mu_2(S_1)$ | 0.0246 | 0.0248 |
| $\mu_3(S_1)$ | 0.0000 | 0.0000 |
| $\mu_4(S_1)$ | 0.0012 | 0.0012 |

### *6.8.6 Equivalence of Algorithms 2 and 4*

Tables 6.8 and 6.9 validate the claim that Algorithm 4 (REPSAMP) is equivalent to Algorithm 2 (REJSAMP) by comparing the first four moments of the slack $S_1$ that is generated by each algorithm for two scenarios. The first row of each table compares the expected value $\mathbb{E}(S_1)$; the remaining rows compare the three central moments $\mu_k(S_1) := \mathbb{E}(S_1 - \mathbb{E}(S_1))^k$, for $k = 2,3,4$.

**Table 6.9:** Equivalence of Sampling Algorithms - 2

| $s = 1$ | $n = 5$ | $d = 0.6$ |
|---|---|---|
| Moment of $S_1$ | REJSAMP | REPSAMP |
| $\mathbb{E}(S_1)$ | 0.1996 | 0.1996 |
| $\mu_2(S_1)$ | 0.0211 | 0.0214 |
| $\mu_3(S_1)$ | 0.0021 | 0.0021 |
| $\mu_4(S_1)$ | 0.0012 | 0.0012 |

Chapter 7

# MAINTAINING A WIRELESS NETWORK AROUND A BOUNDARY - PART II

## 7.1 Summary of Results

In this chapter, we will substantially extend the results of Chapter 6 to more general SCS. We start by considering general IID parents in Section 7.2. Next, we will define the connectivity properties of $\mathscr{G}$ as integrals over $\mathscr{S}$ in Section 7.3. Within Section 7.3, Section 7.3.5 will analyze dynamic attachment and detachment, in which robots attach and detach dynamically from $\mathscr{B}$. We will move on to consider collision-free attachments of finite robots in Section 7.5. In Section 7.4 we show heat maps of the properties of the graph $\mathscr{G}$ computed from Monte Carlo simulations. Finally, Section 7.6 through Section 7.9 detail the complexity of computing $p_{\mathsf{con}}$ for various SCS. The cases of robots with heterogeneous communication radii and inid parents are deferred to those sections, for the reasons explained therein.

## 7.2 IID Coverage by Point Robots

In this section, we consider an SCS driven by a Poisson Point Process (PPP), in which every robot attaches independently to $\mathscr{B}$ following the same spatial parent pdf. In other words, $\mathtt{X}$ consists of iid random variables and defines a PPP on $\mathscr{B}$. Specifically, suppose that the parent pdf and cdf are $h(x)$ and $H(x)$ respectively, both supported on $\mathscr{B}$. Then the

number of points N falling on a subinterval $[a,b]$ of $\mathscr{B}$ is a Poisson random variable with underlying pdf $h(x)$:

$$\mathrm{N}(a,b) \sim \mathrm{Poi}(\lambda) \text{ where } \lambda = H(b) - H(a). \tag{7.1}$$

### 7.2.1 Joint PDF of Positions

We interpret $\mathtt{x}$ and $\mathbf{s}$ as points in $\mathbb{R}^n$ and $\mathbb{R}^{n+1}$, respectively, following the treatment in Section 6.4. The notions of position simplex, slack simplex, connected region $\mathscr{F}$ and disconnected region $\mathscr{U}$ carry over from Section 6.4. The difference now is that the pdf $f_{\mathtt{X}}(\mathtt{x})$ is no longer uniform over $\mathscr{P}$, nor over $\mathscr{S}$. Instead, the parent $h$ generates the joint pdf (David and Nagaraja, 2003)

$$f_{\mathtt{X}}(\mathtt{x}) = n! \prod_{1 \le i \le n} h(\mathtt{x}_i) \mathbf{I}_{\mathscr{P}} \tag{7.2}$$

over the the position simplex, where $\mathbf{I}$ denotes the indicator function as in Equation (6.26). This pdf is called the Janossy pdf (Haenggi, 2012) of the PPP. Changing the argument from $\mathtt{x}$ to $\mathbf{s}$ gives us

$$f_{\mathbf{S}}(\mathbf{s}_{1:n}) = n! \prod_{1 \le i \le n} h(\sum_{j=1}^{i} s_j) \mathbf{I}_{\mathscr{S}} \tag{7.3}$$

Notice that the slacks $S_i$ are not *exchangeable* as was the case with uniform coverage, a fact which makes computations harder. The marginal pdfs of order statistics are determined from Equation (3.13):

$$f_{\mathtt{X}_i}(t) = \sum_{j=i}^{n} \binom{n}{j} F(t)^j (1 - F(t))^{n-j}. \tag{7.4}$$

The marginal pdfs of slacks are more involved. From (David and Nagaraja, 2003), we have

$$f_{S_i}(t) = \frac{n!}{(i-1)!(n-i-1)!} \int_{x=0}^{s} H(x)^{i-1} h(x) h(x+t) (1-H(x+t))^{n-i-1} dx. \tag{7.5}$$

Likewise, the joint pdf of a subset of slacks, and of a subset of order statistics can be determined by repeated marginalization from Equation (7.3).

## 7.3 Properties of $\mathscr{G}$ generated by IID Parents

We will compute the properties of $\mathscr{G}$ in the coming subsections, starting with $p_{\mathsf{con}}$. The formula for finding $p_{\mathsf{con}}$ provides us with a template for partitioning $\mathscr{S}$ into regions that are amenable to computing the following properties of $\mathscr{G}$: the number of connected components of $\mathscr{G}$, the coverage induced by $\mathscr{G}$, and the edge count of $\mathscr{G}$. While these quantities are nontrivial to compute for RGGs of arbitrary dimension (Penrose, 2003), there exist straightforward, if tedious, algorithms to compute them for a single dimension. All these algorithms essentially involve computing the ratio of integrals of the joint pdf $f_{\mathbf{S}}(\mathbf{s})$ over a subset of $\mathscr{S}$.

### *7.3.1 Probability of Connectivity*

The probability of connectivity $p_{\mathsf{con}}$ is the ratio of the volume of the joint pdf lying over $\mathscr{F}$ to that over $\mathscr{S}$:

$$p_{\mathsf{con}} := \frac{\mathrm{Leb}(\mathbf{S}, \mathscr{F})}{\mathrm{Leb}(\mathbf{S}, \mathscr{S})} = \frac{\int_{\mathscr{F}} f_{\mathbf{S}}(\mathbf{s}) d\mathbf{s}}{\int_{\mathscr{S}} f_{\mathbf{S}}(\mathbf{s}) d\mathbf{s}}. \tag{7.6}$$

where $\text{Leb}(\mathbf{S}, \mathscr{S})$ computes the Lebesgue measure of the joint pdf of $\mathbf{S}$ over $\mathscr{S}$. From Equation (6.22), we have that

$$\text{Leb}(\mathbf{S}, \mathscr{U}) = \sum_{\mathbf{v} \in \{0,1\}^{n+1} : 1 \leq \mathbf{1}^T \mathbf{v} \leq n_{\min}} (-1)^{\mathbf{1}^T \mathbf{v}} \text{Leb}(\mathbf{S}, \mathscr{U}(\mathbf{v})). \tag{7.7}$$

As the joint slack pdf $f_{\mathbf{S}}(\mathbf{s})$ does not have exchangeable slacks, Equation (7.7) will require the evaluation of

$$\sum_{i=1}^{n_{\min}} \binom{n}{i} \tag{7.8}$$

integrals in the worst case. We prove in Section 7.7 that finding $p_{\mathsf{con}}$ can be #PH in the worst case.

### 7.3.2 Number of Connected Components

A slack vector $\mathbf{s}$ has a single connected component iff it is connected, i.e if $\mathbf{s} \in \mathscr{F}$. Each unsaturated slack in $\mathbf{s}$ inserts a new connected component into $\mathscr{G}$. Define the component counting function

$$\text{cmp} : \mathscr{S} \mapsto \mathbb{N} \text{ with } \text{cmp}(\mathbf{s}) = \sum_{i=1}^{n+1} \mathbf{I}_{s_i \leq d}, \tag{7.9}$$

where the indicator variable $\mathbf{I}_{s_i \leq d}$ is unity if $s_i$ is connected and zero otherwise, and $\mathbb{N} = \{0, 1, 2, \ldots\}$. By definition, we have $\text{cmp} = n + 1 - \mathbf{1}^T \mathbf{v}$ identically over $\mathscr{U}(\mathbf{v})$. We may then compute the expectation of cmp over $\mathscr{S}$ by writing $\mathscr{S} = \mathscr{U} \cup \mathscr{F}$, and consequently get

$$\mathbb{E}(\text{cmp}) = \text{Leb}(\text{cmp}, \mathscr{S}) = \sum_{\mathbf{v} \in \{0,1\}^{n+1} : \mathbf{1}^T \mathbf{v} \leq n_{\min}} (-1)^{\mathbf{1}^T \mathbf{v}} \int_{\mathscr{U}(\mathbf{v})} (n + 1 - \mathbf{1}^T \mathbf{v}) f_{\mathbf{S}}(\mathbf{s}) d\mathbf{s}. \tag{7.10}$$

### 7.3.3 Coverage Length

To determine the length of $\mathscr{B}$ covered by $\mathbf{s}$, we will introduce the *coverage function* $\mathrm{cov}(\mathbf{s})$. If $\mathbf{s}$ is connected, then its coverage length $\mathrm{cov}(\mathbf{s})$ is the boundary length $s$. If $\mathbf{s}$ has a disconnected slack $s_i$, a length of $s_i - d$ is left without coverage. This motivates us to define cov by

$$\mathrm{cov} : \mathscr{S} \mapsto \mathbb{R} \text{ with } \mathrm{cov}(\mathbf{s}) = s - \sum_{i=1}^{n+1} \max(s_i - d, 0). \tag{7.11}$$

Computing $\mathbb{E}(\mathrm{cov})$ over $\mathscr{S}$ is not simplified by the decomposition $\mathscr{S} := \mathscr{U} \cup \mathscr{F}$, for cov is non-constant over $\mathscr{U}$. A straightforward integration gives us

$$\mathbb{E}(\mathrm{cov}) = \mathrm{Leb}(\mathrm{cov} \cdot \mathbf{S}, \mathscr{S}) = s \cdot \mathrm{Leb}(\mathbf{S}, \mathscr{S}) - \sum_{i=1}^{n+1} \int_{\mathscr{U}} \max(0, s_i - d) f_{\mathbf{S}}(\mathbf{s}) d\mathbf{s}. \tag{7.12}$$

### 7.3.4 Edge Count

We will define the edge counting function $\mathrm{edg}(.)$ over positions rather than slacks. Given the position vector $\mathtt{x}$, there exists an edge between $\mathtt{x}_i$ and $\mathtt{x}_j$ iff $\mathtt{x}_j - \mathtt{x}_i \leq d$. Accordingly, we have

$$\mathrm{edg} : \mathscr{P} \mapsto \mathbb{N}, \text{ with } \mathrm{edg}(\mathtt{x}) := \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (1 - \max(\mathtt{x}_j - \mathtt{x}_i - d, 0)) \tag{7.13}$$

with $\mathbb{E}(\mathrm{edg})$ being the integral of Equation (7.13) over $\mathscr{P}$:

$$\mathbb{E}(\mathrm{edg}) = \mathrm{Leb}(\mathtt{X}, \mathscr{P}) - \int_{\mathscr{P}} \sum_{i,j:1 \leq i < j \leq n} \max(\mathtt{x}_j - \mathtt{x}_i) f_{\mathtt{X}}(\mathtt{x}) d\mathtt{x}. \tag{7.14}$$

### 7.3.5 Dynamic Coverage with IID Attachment

Now we will examine a strategy in which the robot team dynamically attaches and detaches from the boundary, with the pdfs of their attachment being iid on $\mathscr{B}$.

#### Dynamic Attachment

We first consider the case in which robots attach to the boundary without detaching. One robot position is chosen at every time step from the parent pdf $h$ until connectivity is achieved. We compute the expected time until connectivity, or the expected *stopping time* of the SCS. To determine the stopping time, we consider the sequence $(p_i)_{i\in\mathbb{N}}$, where $p_i$ is the probability of connectivity with $i$ robots. Irrespective of the parent pdf $h$, having more robots on $\mathscr{B}$ leads to a greater probability of connectivity. Consequently, the sequence $(p_i)$ is monotonically increasing on the support $[n_{\min},\infty)$ and tends to unity as $i$ grows without bound. We also know that $p_{i:i\leq n_{\min}} = 0$. Consequently, the attachment process will terminate (resp. fail to terminate) at $i \geq n_{\min}$ robots with probability $p_i$ (resp. $1-p_i$). The probability of connectivity being attained by $i$ robots is:

$$\tau_i = \begin{cases} 0 \quad \text{for } 1 \leq i < n_{\min} \\ (1-p_{i-1})p_i \quad \text{for } i \geq n_{\min} \end{cases} \tag{7.15}$$

Thus, $\tau_i$ is a generalized geometric random variable whose probability of success in a trial is distinct from that in its previous one. The expected stopping time is

$$\mathbb{E}(\tau) = \sum_{i=n_{\min}}^{\infty} i\tau_i. \tag{7.16}$$

Since $p_i > p_{n_{\min}}$ for $i > n_{\min}$, we expect connectivity to be achieved more quickly than that of a geometric random variable whose parameter is $p_{n_{\min}}$, so that

$$\mathbb{E}(\tau) \leq \frac{1}{p_{n_{\min}}}. \tag{7.17}$$

**Stopping Time of Connectivity for Uniform parent**

The uniform parent has the special property that $\mathbf{S}$ is jointly uniform over $\mathscr{S}$, with each slack being identically distributed (though not iid) as scaled exponentials of the form $s \cdot \mathrm{Exp}(1)$. Further, the *order statistics* of the slacks, represented by the vector $\mathtt{s}$, formed by sorting $\mathbf{s}$ in increasing order, obey the relations (David and Nagaraja, 2003):

$$\mathbb{E}(\mathtt{S}_i) = \frac{s}{n+1} \sum_{j=1}^{n+1} \frac{1}{j} = \frac{s}{n+1}(H_{n+1} - H_i) \tag{7.18}$$

$$\mathscr{V}(\mathtt{S}_i) = \sum_{j=i}^{n+1} \frac{1}{j^2} \tag{7.19}$$

where $H_n$ denotes the harmonic numbers. The longest slack $\mathtt{S}_{n+1}$ has the expected value $\frac{sH_{n+1}}{n+1}$. To have $\mathtt{s}_{n+1} \leq d$, we need $\frac{H_{n+1}}{n+1} \leq \frac{d}{s}$, which may be solved numerically to get the expected stopping time of connectivity. If $n$ is large we may take $H_n \approx \log n$, providing

$$\frac{\log(n+1)}{n+1} \leq \frac{d}{s} \implies n = \exp\left(-W(\frac{d}{s})\right) - 1, \tag{7.20}$$

where $W$ is the *Lambert W function*.

**Dynamic Attachment and Detachment**

We now consider a scenario in which we require robots to strike a balance between forming a connected network on the boundary and exploring the surrounding environment of the

boundary. Formally, we are given that at every time instant $t \in \mathbb{R}_+$, a robot may be either attached to the load or detached from it; in other words, the robot has a temporal state alphabet $\Sigma := \{A \text{ (attached)}, D \text{ (detached)}\}$.

**Problem Statement.** *Design the robots' rates of switching between states, with a guarantee on the expected amount of time that the boundary will have a connected network.*

To analyze the behavior of the robots, we introduce the temporal state $\mathrm{N}(t) = \begin{bmatrix} N_A(t) & N_D(t) \end{bmatrix}^T$, whose entries denote the number of robots in states $A$ and $D$, respectively. We assume that the total number of robots is conserved, which implies that

$$N_A(t) + N_D(t) = N_A(0) + N_D(0). \tag{7.21}$$

Now we suppose that robots change state according to the chemical reactions

$$A \xrightarrow{r_{AD}} D \text{ and } D \xrightarrow{r_{DA}} A \tag{7.22}$$

where $r_{ij}$, the reaction rate constant, is the probability per unit time of a robot in state $i$ to switch to state $j$. The populations of robots in both states evolve over time as

$$\frac{d}{dt}\mathrm{N}(t) = \begin{bmatrix} -r_{AD} & r_{DA} \\ r_{AD} & -r_{DA} \end{bmatrix} \mathrm{N}(t). \tag{7.23}$$

At equilibrium, $\frac{d}{dt}\mathrm{N}(t) = \mathbf{0}$, and so Equation (7.23) yields

$$\frac{N_A^*}{N_D^*} = \frac{r_{DA}}{r_{AD}}. \tag{7.24}$$

We can solve for $N_A*$ and $N_D*$ using Equation (7.21) and Equation (7.24). By designing $r_{DA}$ to be sufficiently large, we thereby make $N_A^*$ correspondingly large, and thus guarantee the expected time for which $\mathscr{B}$ will be connected.

## 7.4 Properties of $\mathscr{G}$ Computed by Monte-Carlo Sampling

This section computes the properties of $\mathscr{G}$ generated by averaging the results of 2500 Monte Carlo trials, for point robots. In each trial, a configuration of robots was sampled on $\mathscr{B}$ using the parent pdf, and the graph $\mathscr{G}$ was computed using the robot positions `x` and the communication range $d$. Each of the properties of $\mathscr{G}$ described in Section 7.3 was then computed, and then averaged across trials to get their expected value. These results can be used by a system designer to select the number of robots and the robot communication radius, that will yield desired boundary coverage properties in expectation.

### *7.4.1 Uniform parent*

Figure 7.1 and Figure 7.2 show the three-dimensional contour plots of four graph properties: probability of connectivity, number of components, mean vertex degree, and covered length as a function of $n$ and $d$, for the uniform parent on a boundary with length $s = 15$. The robots are considered to be point robots.

### *7.4.2 Piecewise Uniform Parent*

Figure 7.3 and Figure 7.4 repeat the above plots for a piece-wise uniform parent on the same boundary. We define a $k$-piecewise uniform (PWU) parent to be one whose pdf is the

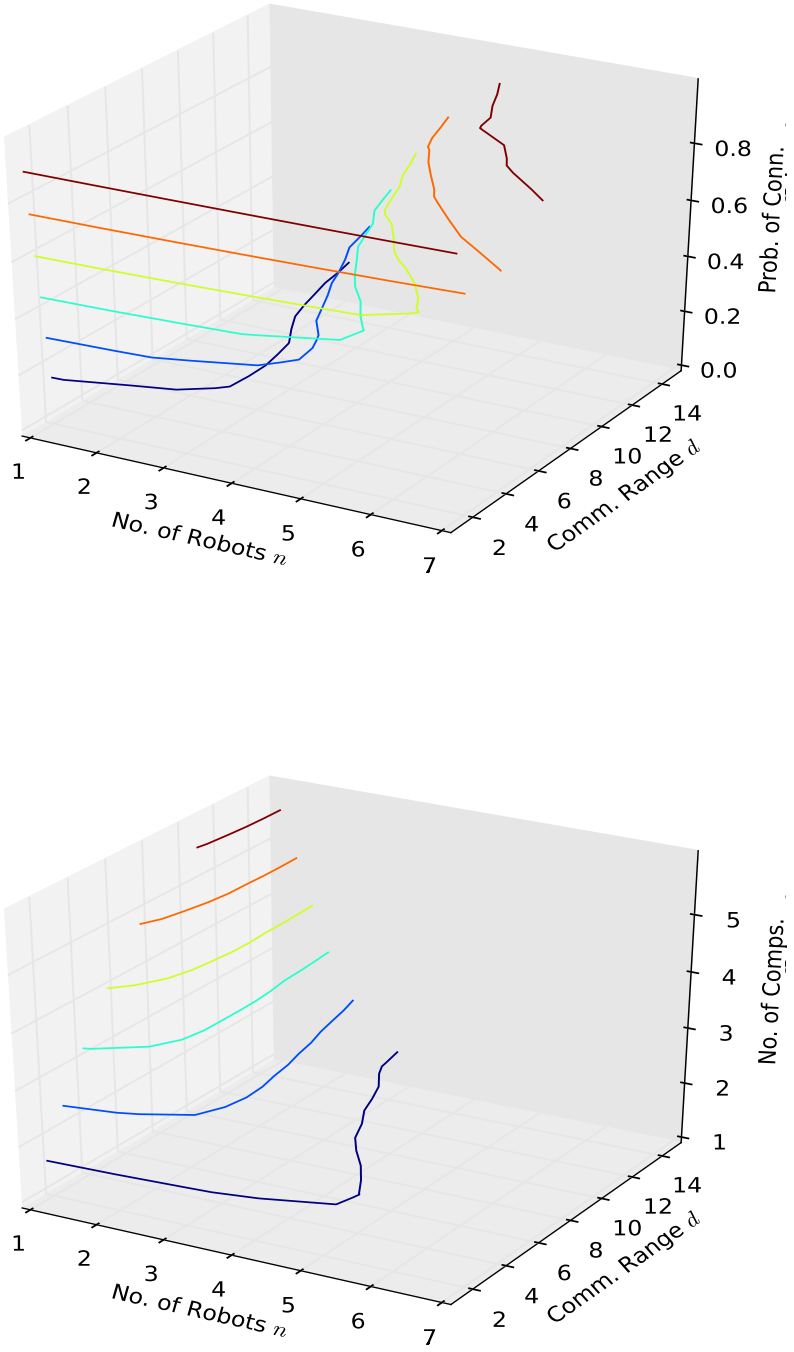



**Fig. 7.1:** Properties of $\mathscr{G}$ Generated by Uniform Parent: Contour plots of $p_{\text{con}}$ (top) and mean number of components (bottom)

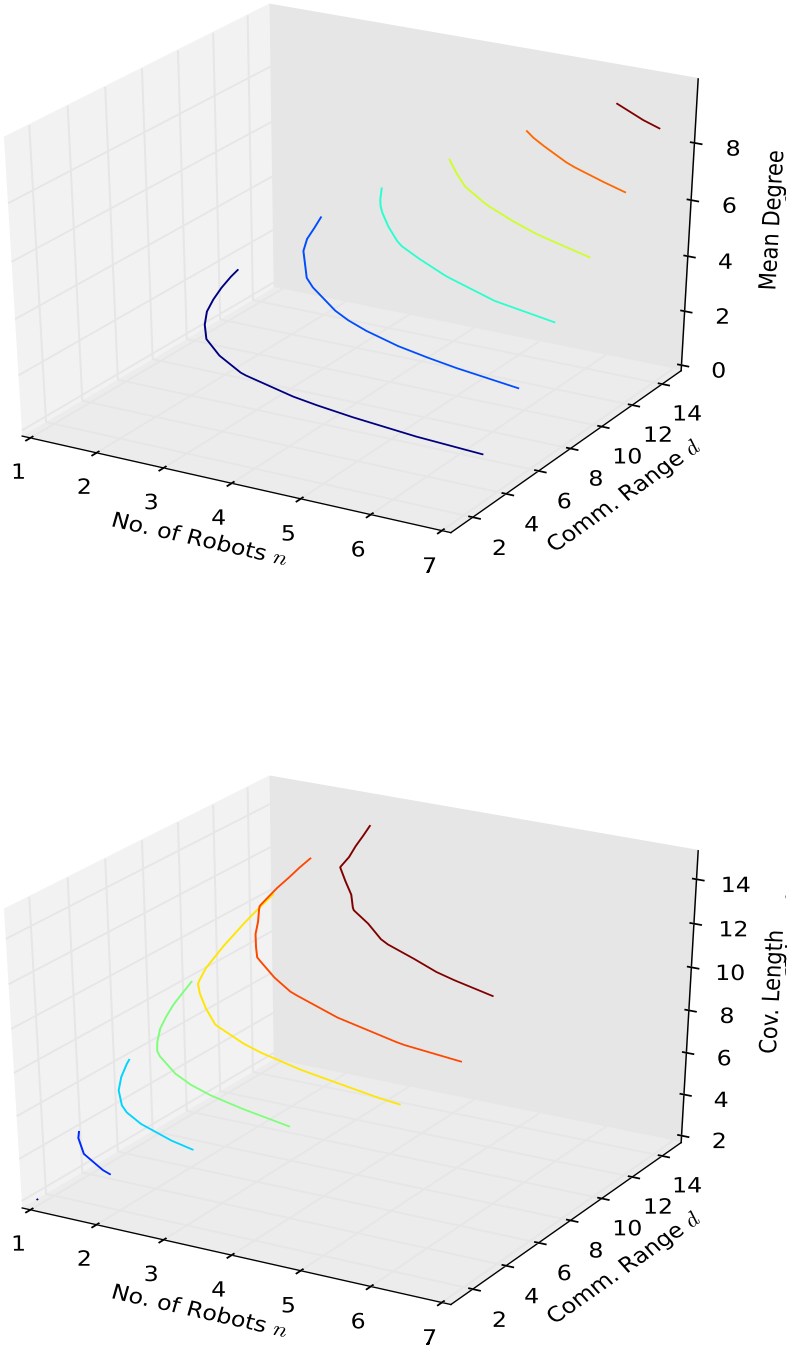



**Fig. 7.2:** Properties of $\mathscr{G}$ Generated by Uniform Parent: Contour plots of mean vertex degree (top) and mean covered length (bottom)

union of $k$ uniform pdfs on its support. Our parent of interest is the 3-PWU pdf:

$$f(x) = \begin{cases} 0.2, \text{ for } x \in [0,5] \\ 0.6, \text{ for } x \in [5,10] \\ 0.2, \text{ for } x \in [10,15]. \end{cases} \tag{7.25}$$

Note that Figure 7.3 and Figure 7.4 closely resemble the plots Figure 7.1 and Figure 7.2 for the uniform parent. This resemblance occurs for other parent pdfs that we tried: Beta, truncated normal, and triangular pdfs on $\mathscr{B}$.

## 7.5 Uniform Coverage by Finite Robots

We now consider SCS with *finite robots*, each of which has a nonzero diameter $R$. Unlike the case of point robots, the *maximum number* of attached robots is finite and given by

$$n_{\max} = \lfloor \frac{s}{R} \rfloor. \tag{7.26}$$

We say that two robots conflict with each other if they physically overlap, just as in Section 3.7.3, and *conflict-free* (CF) otherwise. We say that a configuration of robots on $\mathscr{B}$ is CF iff it has no pairs of conflicting robots. CF positions are a realization of a *Matern hard-core PP* (Haenggi, 2012), which prohibits its points from lying within a threshold distance of each other. The valid range for $n$ is $[n_{\min}, n_{\max}]$. The case $d = R$ is of special interest to us, since it is an instance of Renyi's Parking Problem.

**Problem Statement.** *Renyi's Parking Problem (Sikiric and Itoh, 2011; Dvoretzky and Robbins, 1964) Cars of unit length park uniformly randomly on a segment of length s,*

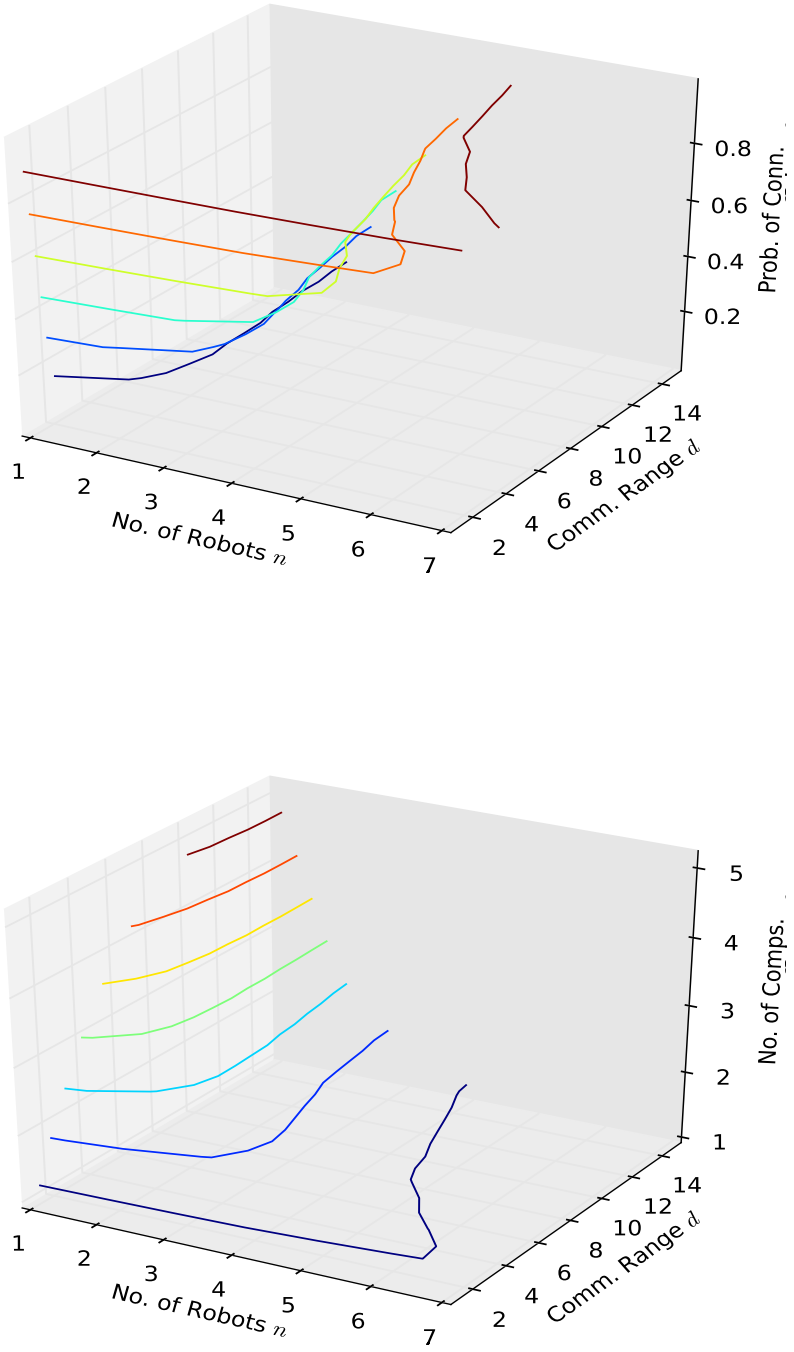



**Fig. 7.3:** Properties of $\mathscr{G}$ Generated by 3-PWU Parent: Contour plots of $p_{\text{con}}$ (top) and mean number of components (bottom)

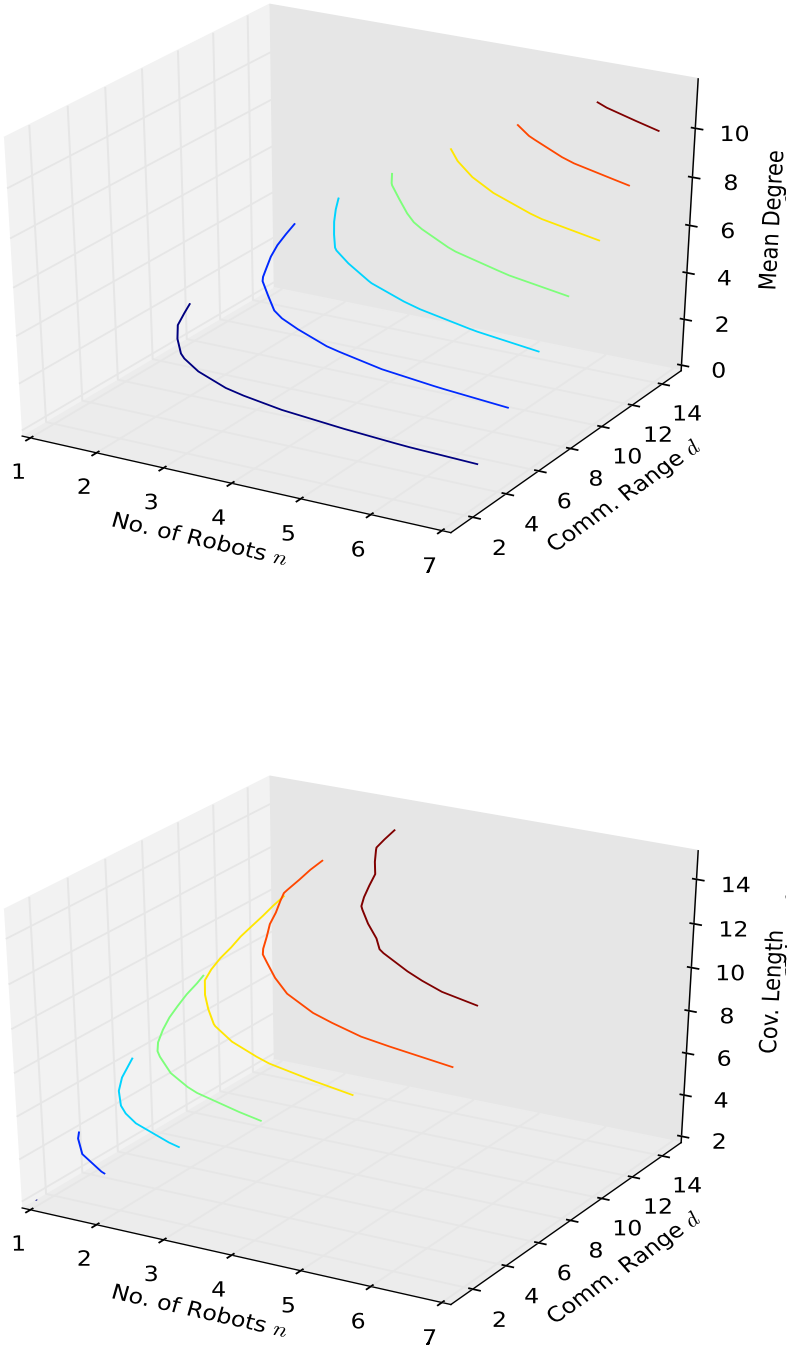



**Fig. 7.4:** Properties of $\mathscr{G}$ Generated by 3-PWU Parent: Contour plots of mean vertex degree (top) and mean length covered (bottom)

*avoiding conflicts, until no parking space is available for the next car. Analyze the pdf of the final number of parked cars,* N.

The mean number of parked cars, $\mathbb{E}(\mathrm{N})$, obeys a delay integral equation with the asymptotic solution

$$\lim_{s\to\infty} \mathbb{E}(\mathrm{N}) = n_{pc} \cdot s \approx 0.748s, \tag{7.27}$$

where $n_{pc}$ is Renyi's parking constant (Rényi, 1958). This result implies that we expect 75% of the segment to be occupied by cars at the point where there is no more room to accommodate another car. An exact solution for $\mathbb{E}(\mathrm{N})$ leads to an intractable $\lfloor s \rfloor$-dimensional integral (Muller, 2007). Our SCS with fixed $n$ and uniformly random attachments is a special case of Renyi's Parking Problem in which N is trivial to compute. However, to our knowledge, there has been no analysis of the spatial pdfs that are generated by the parked cars in this problem, which we provide in Section 7.5.1.

### 7.5.1 Connectivity of Conflict-Free Parking

We now formulate the CF equivalent of the geometric form of the point-robot attachment scenario introduced in Section 6.3. Define the *position* of a robot as that of its left end, so that robot $\mathscr{R}_i$ located at $x_i$ occupies the interval $[x_i,\ x_i + R]$. The support of all attached robot positions is $\mathscr{B}' = [0, s-R]$, which ensures that no robot extends beyond the boundary endpoint $x = s$. We introduce two artificial robots at $\mathrm{x}_0 = -R$ and $\mathrm{x}_{n+1} = s$. It is easy to see that $\mathrm{x}$ is CF iff

$$0 \le \mathrm{x}_0, \quad \mathrm{x}_n \le s - R, \ \text{ and } \mathrm{x}_i - \mathrm{x}_{i-1} \ge R, \text{ for } i = 1 \ldots n, \tag{7.28}$$

and define $\mathscr{P}_{\text{CF}}$ to be the set of CF position vectors. Likewise define the CF subset of $\mathscr{S}$ and the resulting connected region by

$$\mathscr{S}_{\text{CF}} := \{\mathbf{s} \in \mathscr{S} : R \cdot \mathbf{1}^T \leq \mathbf{s}\} \tag{7.29}$$

$$\mathscr{F}_{\text{CF}} := \mathscr{S}_{\text{CF}} \cap \mathscr{H} = \{\mathbf{s} \in \mathscr{S} : R \cdot \mathbf{1}^T \leq \mathbf{s} \leq d \cdot \mathbf{1}^T\}. \tag{7.30}$$

Geometrically, $\mathscr{S}_{\text{CF}}$ is a simplex with the hypercuboids $\mathbf{0} \leq s_i \leq R$ removed. Reasoning as in Section 6.4, we have $p_{\mathsf{con}}(\mathscr{G}) = \text{Vol}(\mathscr{F}_{\text{CF}})/\text{Vol}(\mathscr{S}_{\text{CF}})$; however, we are unable to find a closed form expression for $p_{\mathsf{con}}$ as we did there. The lack of a simplifying expression for $p_{\mathsf{con}}$ means that the computation of $p_{\mathsf{con}}(\mathscr{G})$ has to involve the triangulation of $\mathscr{F}_{\text{CF}}$ into simplices, a time-consuming operation that we explicitly avoided in Equation (6.18). Likewise, expressions for the order statistics and slacks of CF positions are obtained by integrating the uniform joint pdf over $\mathscr{S}_{\text{CF}}$ instead of $\mathscr{S}$, as are the formulas for the properties of $\mathscr{G}$ derived in Section 7.3.

We conclude this chapter by investigating the complexity of computing $p_{\mathsf{con}}$ for various problem scenarios. While these computations are not required by the problem statement in Section 6.1.2, they nonetheless provide insight into the problem scenarios that lead to highly symmetric $\mathscr{F}$ versus those that do not. As a rule of thumb, computing $p_{\mathsf{con}}$ is easy for the former cases, but hard for the latter. We will formally derive lower bounds for computing $p_{\mathsf{con}}$ using Theorem 3.6.1.

We now investigate the complexity of exactly computing $p_{\mathsf{con}}$ using Equation (6.22). We begin with a pseudo-P lower bound for computing $p_{\mathsf{con}}$ for the uniform parent. We will then discuss lower bounds for non-uniform parents. Define the complexity theoretic problem $\mathsf{PCON}(h,s,d,n) \mapsto p_{\mathsf{con}}$, with

Input: Parameters of SCS : encoding of $h$ ; $s,d \in \mathbb{Q}_+$ ; $n \in \mathbb{N}$

Output: Probability of connectivity $p_{\mathsf{con}} \in \mathbb{Q}_+$

Rational inputs and outputs are specified as exact reduced fractions; for example, $s$ is input as the pair $(\mathrm{num}(s), \mathrm{den}(s))$. Let $\mathsf{PCON}(\mathrm{U})$ denote the subproblem of $\mathsf{PCON}$ having a uniform parent.

**Theorem 7.6.1.** $\mathsf{PCON}(\mathrm{U})$ *can be solved in* $\Omega(n)$ *and* $O(n \log n)$ *time.*

*Proof. Upper bound:* Equation (6.16) is a possible solution for $\mathsf{PCON}(\mathrm{U})$; therefore, the worst-case running time of an algorithm implementing it constitutes an upper bound for $\mathsf{PCON}(\mathrm{U})$. This algorithm uses binary exponentiation compute terms of the form $(1 - kd/s)^n$ on the right side of Equation (6.16) to run in $O(n_{\min} \log n)$ time; its worst-case instances, which have $n > n_{\min}$ take $O(n \log n)$ time, which forms an upper bound for any solution to $\mathsf{PCON}$. This algorithm thus solves $\mathsf{PCON}$ in pseudo-P time.

*Lower bound:* Consider $\mathsf{PCON}(\mathrm{U})$ instances with $s = \frac{4}{3}d$, for which

$$\mathsf{PCON} = 1 - (n+1) \cdot \frac{1}{4^n} \approx 1 - \frac{n}{4^n} \tag{7.31}$$

The input size for these instances is $O(\log n)$, but the output size is exponential in the input size, implying an $\Omega(n)$ lower bound for any algorithm for $\mathsf{PCON}(\mathrm{U})$. The upper bound of $O(n \log n)$ exceeds the lower bound $\Omega(n)$ only by a polynomial in the input size of $\mathsf{PCON}(\mathrm{U})$. □

### 7.6.1 *Computing Vol($\mathscr{F}$) and* Vol($\mathscr{U}$) *for Uniform Parents*

The $\sqrt{n}$ in the formula for $\mathrm{Vol}(\mathscr{U})$ obtained from Equation (6.20) makes it impossible to provide a bounded decimal expansion to $\mathrm{Vol}(\mathscr{U})$, and to $\mathrm{Vol}(\mathscr{F})$ as well. Hence the complexity of writing down $\mathrm{Vol}(\mathscr{F})$ is *infinite*, except when $n$ is the square of an integer. Remedying this unbounded expansion requires us to leave the term $n!/\sqrt{n+1}$ unsimplified. As a result, the bounds for $\mathsf{PCON}(\mathrm{U})$ apply to $\mathrm{Vol}(\mathscr{F})$.

Now note that $\mathscr{U}(\mathbf{v})$ for any $\mathbf{v} \in \{0,1\}^{n+1}$ involves the intersection of $\mathscr{S}$ with a hypercuboid $\mathscr{C}'$ formed by the product of intervals $\mathscr{C}'_i$, where

$$\mathscr{C}'_i := \begin{cases} [0,s] \text{ if } v_i = 0 \\ [0,d] \text{ otherwise.} \end{cases} \tag{7.32}$$

The sides of $\mathscr{C}'$ thus take on only two possible values; consequently, $\mathscr{C}'$ is too specific for us to apply the hardness result in Theorem 3.6.1 to $\mathrm{Vol}(\mathscr{U}(\mathbf{v}))$. It follows that this volume can be computed in pseudo-P time. This pseudo-P time bound also applies to all the connectivity properties of $\mathscr{G}$ listed in Section 7.3.

## 7.7 PCON for Nonuniform IID parents

Nonuniform iid parents arise in a variety of problem scenarios, for example:

1. Consider a robot team that wishes to gather say at must gather at a target location $x \in \mathscr{B}$, and neglect collisions among them. If the robots have noisy odometry, then the robots' parent will be a truncated Gaussian supported on $\mathscr{B}$. This truncated Gaussian parent has mean $x$ and has fixed variance $\sigma^2$, determined by the uncertainty in the robots' wheel odometry.

2. Consider the case where $\mathscr{B}$ is the perimeter of a payload with a nonuniform distribution of mass, with one side of the load heavier than the other. More robots are needed to support the heavier side during transport, and so the parent pdf will have a higher density along the boundary of that side.

3. Consider the adsorption of nanoparticles on a heterogeneous boundary, that is comprised of several distinct subintervals, each with a different adsorption rate. Then the parent of each particle is a mixture of pdfs that are supported on each subinterval.

We will now give an example of a nonuniform parent whose $p_{\mathsf{con}}$ is #PH to compute. For this purpose, define a $k$-piecewise uniform ($k$-PWU) pdf over a finite support $[0,L]$ as follows. Partition the support into $k$ nonempty subintervals as

$$[0,L] := [0,L_1] \cup [L_1,L_2] \cup \ldots \cup [L_k,s], \tag{7.33}$$

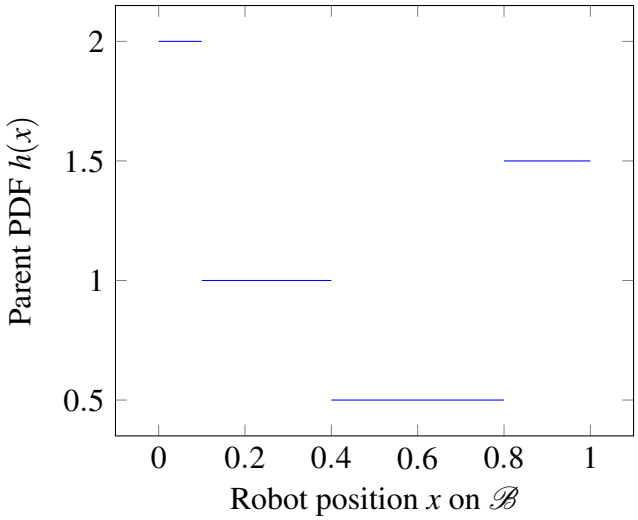



**Fig. 7.5:** 4-Uniform PDF on the boundary $\mathscr{B} := [0,1]$. When the number of pieces is not constant, computing PCON becomes #PH.

On the $i$-th subinterval, $f(x)$ is defined to be the constant $p_i \in [0,1]$, which are chosen to satisfy $\sum p_i(L_i - L_{i-1}) = 1$, and consequently $f$ is a pdf on $\mathscr{B}$. We define an $n$U pdf to be one which has $n$ pieces, on each of which it is constant. A 4U pdf is shown in Figure 7.5.

**Theorem 7.7.1.** PCON($n$U) *is* #PH.

*Proof.* Given the VHSP instance with the dimension $n+1$, having the hypercuboid $\mathscr{C}' := \prod_{1 \le i \le n+1}[0, l_i]$ and the half-space $\mathscr{T} := \{\mathbf{s} \in \mathbb{R}^n : \mathbf{1}^T\mathbf{s} \le b\}$. Let $L := \sum l_i$. Define the equivalent instance of PCON to have the parameters

$$s := \frac{b}{L},\ d := 1\ \text{, and } h := \begin{cases} \frac{l_i}{L} \text{ if } x \in [i, i+1] \\ 0 \text{ otherwise} \end{cases}\ \text{.,} \tag{7.34}$$

Define $Y_i \sim H(X_i)$ to be the probability integral transform of $X_i$. From the definition of $Y_i$, we have that $\mathbb{P}(x_i \in [i, i+1]) = \mathbb{P}(y_i \in [0, \frac{l_i}{L})]$. It follows that if X is connected, then $\mathbf{Y}$ lies within $\mathscr{C}'$. Moreover, $\mathbf{Y}$ is jointly uniform on the half-space

$$\mathscr{T}' = \{\mathbf{y} \in \mathbb{R}^n : \mathbf{y} \ge 0 \text{ and } \sum y_i \le s\}. \tag{7.35}$$

Thus, $p_{\mathsf{con}} = \mathrm{Vol}(\mathscr{T}' \cap \mathscr{C}')/\mathrm{Vol}(\mathscr{C}')$. Since we can compute $p_{\mathsf{con}}$ in P time, we can solve VHSP by returning $p_{\mathsf{con}} \cdot \mathrm{Vol}(\mathscr{C}')$ in P time. □

### 7.7.1 *More General Parents whose* PCON *is* #PH

We may extend Theorem 7.7.1 to more general parents that satisfy the constraints:

$$\int_{(i-1)d}^{d} h(x)dx = l_i \text{ for all } i = 1,\ldots,n, \text{ where } l_i \geq 0 \text{ and} \sum l_i = 1. \tag{7.36}$$

Since Equation (7.36) provides us with $n$ constraint equations, $h$ needs to contain at least $n$ parameters to fit them, e.g. polynomials of degree $n$, with arbitrary coefficients. More generally, if $f_1,\ldots,f_n$ are arbitrary pdfs with unit supports, each having at least one parameter, then their mixture

$$h(x) = f_i(x) \text{ if } x \in [i-1,i)\mathbf{I}_{\mathscr{B}}, \text{ where } \mathscr{B} = [0,n+1] \tag{7.37}$$

can obey Equation (7.36). Consequently, computing PCON for this mixture is #PH.

For nonuniform parents such as truncated Gaussians and Beta pdfs, PCON could either be pseudo-P or #PH. We do not currently have reductions for these parents, nor can we derive a pseudo-P algorithm for computing $p_{\mathsf{con}}$. However, we would like to state a rule of thumb: the more arbitrary parameters (such as coefficients) a parent has, the more likely it is to have an instance that satisfies Equation (7.36), and as such its PCON is more likely to be #PH.

We now consider a *heterogeneous* robot team whose Wi-fi adapters have different levels of transmission power, so that $\mathscr{R}_i$ has communication range $d_i$. We call $\mathscr{R}_i$ *weaker* (resp. *stronger*) than $\mathscr{R}_j$ iff $d_i < d_j$ (resp. $d_i > d_j$). If $d_i = d_j$, we say that the two robots have equal power. Then the robot network is represented by the digraph $\mathscr{G}$, whose directed edges are of the form $i \to j$ iff $\mathscr{R}_i$ can transmit to $\mathscr{R}_j$. In general, edges are not bidirectional, since a weaker robot will not sense a stronger one, even though the converse is true. Without losing generality, we assume that $d_i \leq d_j$ iff $i < j$.

Consider a configuration in which the robots are arranged from left to right in increasing order of their ids, as shown in Figure 7.6, wherein $\mathscr{R}_i$ takes up position $\mathtt{x}_i$. For each robot to be able to communicate to its neighbours, the configuration needs to satisfy the $n+1$ constraints:

$$s_1, s_2 \leq d_1;\ s_3, s_4 \leq d_2;\ \ldots\ ; \begin{cases} s_n, s_{n+1} \leq d_{(n+1)/2} \text{ if } n \text{ is odd} \\ s_{n-1}, s_n \leq d_{n/2} \text{ and } s_{n+1} \leq d_{n/2+1} \text{ if } n \text{ is even} \end{cases} \tag{7.38}$$

or equivalently,

$$s_i \leq d_{\lfloor \frac{i+1}{2} \rfloor}, \text{ for } i = 1, \ldots, n. \tag{7.39}$$

We call such a configuration a *1-hop connected* configuration, since every robot is just one hop away from adjacent ones.

**Note** 1-hop connectivity is different from connectivity. For example, in Figure 7.6, if $\mathscr{R}_2$ is too weak to communicate with $\mathscr{R}_3$, it may route its messages destined for $\mathscr{R}_3$ through $\mathscr{R}_1$.

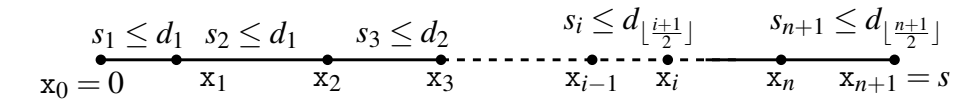



**Fig. 7.6:** A 1-hop connected configuration of $n$ robots with distinct communication ranges requires slacks to obey $s_i \le d_j$, where $j = \lfloor \frac{i+1}{2} \rfloor$. We assume that the robots attach in the same order as their indices, i.e. $\mathscr{R}_i$ with range $d_i$, occupies position $\mathrm{x}_i$.

We do not have a hardness result for connectivity, and will restrict ourselves to the more restricted case of 1-hop connectivity throughout this section.

For all $n$, the bounds for the slacks generated by a configuration of $n$ robots consist of the smallest communication ranges, twice repeated. If we do not make the assumption that robots attach in order of their indices, we find that a connected configuration will have two adjacent slacks $s_j, s_{j+1}$ not exceeding $d_i$ for every $i : 1, \ldots, n/2+1$; if $n$ is even, one remaining slack will not exceed $d_{n/2+1}$. To make these notions more precise, we define the *dimension multiset* of a team of robots to be

$$\mathrm{dim} := \begin{cases} \{d_1, d_1, \ldots, d_{\frac{n+1}{2}}, d_{\frac{n+1}{2}}\} \text{ if } n \text{ is odd,} \\ \{d_1, d_1, \ldots, d_{\frac{n}{2}}, d_{\frac{n}{2}}, d_{n/2+1}\} \text{ if } n \text{ is even.} \end{cases} \tag{7.40}$$

Further, we define a *canonical permutation* of the dimension multiset as one which places equal elements adjacently. A canonical permutation of dim corresponds to a connected configuration of robots which attach in an arbitrary order, and not necessarily in the order of their ids.

**Example 7.8.1.** *Suppose that two robots $\mathscr{R}_1$ and $\mathscr{R}_2$ with ranges $d_1$ and $d_2 : d_2 >= d_1$ attach to $\mathscr{B}$. Their dimension multiset is* $\mathrm{dim} = d_1, d_1, d_2$. *The canonical permutations of* dim *are*

$$\mathrm{Perm}(\mathrm{dim}) := \{(d_1, d_1, d_2), (d_2, d_1, d_1)\}, \tag{7.41}$$

*and the number of canonical permutations is* $|\text{Perm}(\text{dim})| = 2$. *The first permutation* $(d_1, d_1, d_2)$ *tells us that, if the position of* $\mathscr{R}_1$ *comes before that of* $\mathscr{R}_2$ *(i.e.* $x_1 \leq x_2$*), then a connected configuration will have* $s_1 \leq d_1, s_2 \leq d_1, s_3 \leq d_2$. *In other words, its slack vector lies in the cuboid* $[0,d_1] \times [0,d_1] \times [0,d_2]$. *Likewise, the second permutation tells us that if* $x_2 \leq x_1$*, then the slack vector of a connected configuration will lie in the cuboid* $[0,d_2] \times [0,d_1] \times [0,d_1]$.

*When a third robot* $\mathscr{R}_3$ *attaches,* dim *becomes* $\{d_1, d_1, d_2, d_2\}$ *and has the canonical permutations*

$$\text{Perm}(\text{dim}) := \{(d_1, d_1, d_2, d_2), (d_2, d_2, d_1, d_1)\}, \tag{7.42}$$

*with* $|\text{Perm}(\text{dim})| = 2$.

Every permutation perm in Perm(dim) defines an induced hypercuboid $\mathscr{H}_{\text{perm}}$ that a slack vector must lie in to be connected. Thus, for a slack vector to be 1-hop connected, it needs to lie in the union of hypercuboids induced by Perm,

$$\mathscr{H}_{:} = \bigcup_{\text{perm} \in \text{Perm}(\text{dim})} \mathscr{H}_{\text{perm}}. \tag{7.43}$$

Consequently, the 1-hop connected region is $\mathscr{F} := \mathscr{S} \cap \mathscr{H}$.

**Example 7.8.2.** *Suppose as in Example 7.8.1 that robots* $\mathscr{R}_1$ *and* $\mathscr{R}_2$ *attach to* $\mathscr{B}$. *The 1-hop connectivity hypercuboids are*

$$\mathscr{H}_1 := [0,d_1] \times [0,d_1] \times [0,d_2], \tag{7.44}$$

$$\mathscr{H}_2 := [0,d_2] \times [0,d_1] \times [0,d_1]. \tag{7.45}$$

*The union of the two cuboids is $\mathscr{H} := \mathscr{H}_1 \cup \mathscr{H}_2$, and the 1-hop connected region is $\mathscr{F} = \mathscr{S} \cap \mathscr{H}$. Note that the hypercuboids' intersection is the cube $\mathscr{C} = [0, d_1]^3$.*

*Note that s falls into 3 ranges. (a) If $s \leq 3d_1$, then $\mathscr{S}$ overlaps with the cube $\mathscr{C}$, and we are back to the case of equal communication ranges addressed in Section 7.6. (b) If s satisfies $3d_1 < s < 2d_1 + d_2$, then $\mathscr{S}$ does not intersect $\mathscr{C}$ at all. Instead, its intersection with $\mathscr{H}$ is the disjoint union of its intersections with $\mathscr{H}_1$ and $\mathscr{H}_2$. (c) If $s > 2d_1 + d_2$, then $\mathscr{F} = \varnothing$ and $p_{\mathsf{con}} = 0$.*

For our reduction, we will be most interested in the case in which $\mathscr{S}$ has disjoint overlaps with each 1-hop connectivity hypercuboid, i.e. we have

$$\mathscr{S} \cap \mathscr{H}_i \cap \mathscr{H}_j = \varnothing, \text{ but } \mathscr{S} \cap \mathscr{H}_i \neq \varnothing \tag{7.46}$$

where $\mathscr{H}_i$ and $\mathscr{H}_j$ are distinct 1-hop connectivity hypercuboids induced by $\mathrm{Perm}(\mathrm{dim})$. Continuing the reasoning in Example 7.8.2, we then need to have $s$ sufficiently large such that $\mathscr{S}$ does not have nonempty intersections with $\mathscr{H}_i \cap \mathscr{H}_j$. At the same time, $s$ needs to be sufficiently small such that $\mathscr{S}$ has a nonempty intersection with each $\mathscr{H}_i$. Let $\max(\mathrm{dim})$ denote the maximum entry, or one of the two possible maximal entries, in dim ; let $\sum \mathrm{dim}$ denote the sum of all entries in dim, counting repetitions. It is then easy to see that when $s$ lies in the range $(\sum \mathrm{dim} - \max(\mathrm{dim}), \sum \mathrm{dim})$, it meets the two overlap requirements in Equation (7.46). We call this interval the *maximal interval* for the communication ranges $d_{1:n}$.

Given the communication ranges $d_{1:n}$, pick an arbitrary $s$ from its maximal interval. From the above reasoning, we have in addition that

$$\text{Vol}(\mathscr{F}) = |\text{Perm}(\text{dim})|\text{Vol}(\mathscr{S} \cap \mathscr{H}_1), \tag{7.47}$$

where $\mathscr{H}_1$ is one of the hypercubes induced by Perm. It follows that

$$p_{\mathsf{con}} = |\text{Perm}(\text{dim})|\frac{\text{Vol}(\mathscr{S} \cap \mathscr{H}_1)}{\text{Vol}(\mathscr{S})}. \tag{7.48}$$

We define $\mathsf{PCON}-1\text{HOP}$ as the heterogeneous single-hop analog of $\mathsf{PCON}$:

Input: $s, h, n$ as in $\mathsf{PCON}$ ; communication ranges $d_{1:n}$, all of which are positive rationals

Output: Probability of 1-hop connectivity, $p_{\mathsf{con}} \in \mathbb{Q}$

**Theorem 7.8.1.** $\mathsf{PCON}-1\text{HOP}$ *is* $\#\mathsf{PH}$ *for uniform parents.*

*Proof.* We will reduce to $\mathsf{PCON}-1\text{HOP}$ from a $\#\mathsf{PH}$ subset of $\mathsf{VHSP}$ that has *twice repeated* hypercuboid dimensions. Consider the instance of $\mathsf{VHSP}$ that asks for the volume of overlap between the hypercuboid $\mathscr{C}' \in \mathbb{R}^n$ defined by

$$\mathscr{C}' := [0, l_1]^2 \times [0, l_2]^2 \times \ldots \times [0, l_{n/2}]^2, \tag{7.49}$$

where $n$ is even, and the half-space

$$\mathscr{T}' := \{\mathbf{s} \in \mathbb{R}^n : \mathbf{1}^T\mathbf{s} \leq b\}. \tag{7.50}$$

We will assume that $b$ lies in the maximal interval for the dimensions $l_{1:n/2}$. Note that this assumption merely makes the $\mathsf{VHSP}$ instance harder, since a value of $b$ lying in a nonmaximal interval will lead $\mathscr{T}'$ to intersect with a hypercuboid which has more dimensions equal

to one another than $\mathscr{C}'$. Further, the set of such instances of VHSP with twice repeated dimensions is #PH, following the same reduction from KNAPSACK given in (Dyer and Frieze, 1988). Also, we may assume without loss of generality that $l_i \leq l_{i+1}$.

Now we will set up an equivalent instance of PCON for robots of distinct ranges. This instance has $n-1$ robots attaching to a boundary of length $s := b$, creating $n$ slacks in all. The communication range of $\mathscr{R}_i$ is $d_i := l_{\min(i,n/2)}$. The dimension of each hypercuboid induced by Perm(dim) matches that of $\mathscr{C}'$. Likewise, the slack simplex $\mathscr{S}$ is that subset of $\mathscr{T}'$ lying in the positive orthant of $\mathbb{R}^n$. As such, the solution to VHSP is given by $p_{\mathsf{con}}/|\text{Perm}(\text{dim})|$, which is computable in P time if $p_{\mathsf{con}}$ is known.

□

It is immediately clear that finding $\text{Vol}(\mathscr{F})$ and $\text{Vol}(\mathscr{U})$ is #PH for connectivity with distinct communication ranges. With robots of identical communication range, $\mathscr{F}$ was more symmetric compared to its counterpart with distinct communication ranges. Exploiting this symmetry led to relatively short formulas for PCON and $\text{Vol}(\mathscr{F})$. On the other hand a swarm with distinct ranges is sufficiently general that its connected region can be the interesection of an arbitrary half-space with the unit hypercube. The tradeoff for this expressiveness is an increase in complexity of the connectivity problems. Computing $p_{\mathsf{con}}$ has a Fully Polynomial Randomized Approximation Scheme (FPRAS), which samples a uniform pdf over a subset of $\mathscr{F}$ in P time, using the Markov Chain Monte Carlo (MCMC) method (Jerrum, 2003). Combining MCMC with Inverse CDF Sampling enables us to sample arbitrary IID pdfs over $\mathscr{F}$. This approach is sufficiently general that it can be adapted

to arbitrary joint pdfs over $\mathscr{F}$, in which case it becomes the Metropolis-Hastings sampling algorithm (Chib and Greenberg, 1995).

## 7.9 PCON for INID Parents

Suppose we consider the adsorption of several *types* of particles on $\mathscr{B}$, where each type of particle has a different adsorption profile. Such a problem scenario leads to a set of independent but non-identically distributed parents, i.e. inid parents. Likewise, when a team of robots with noisy odometry attaches to $\mathscr{B}$, with each robot having a different noise profile, their positions on the boundary are governed by inid normal parent pdfs. An example of this scenario is shown in Figure 7.7. The formulas for order statistics of inid parents are nontrivial (David and Nagaraja, 2003, Ch.5). Finding the joint pdfs of order statistics and slacks involve the use Theorem 3.4.1, and consequently entails the computation of the permanent of the CDF matrix $\mathbb{L}$ of the parents. We currently have no concrete hardness results for the computation of these pdfs, nor for the associated PCON problem. Since these proofs are likely to be quite lengthy, we plan to supply them in our future work (Kumar and Berman, 2016a).

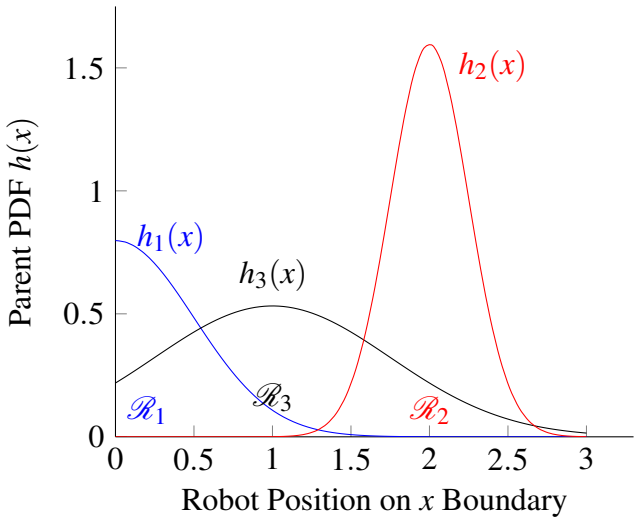



**Fig. 7.7:** Robots $\mathscr{R}_1, \mathscr{R}_2$ and $\mathscr{R}_3$ attaching to $\mathscr{B}$ with three distinct truncated Gaussian noise profiles having INID parents $\mathbf{h} = (h_1, h_2, h_3)$.

Chapter 8

# FUTURE SIMULATIONS

In this chapter, we outline the challenges to answering Part 5 of the main problem. We are currently unable to merge together state-of-the-art robot simulators such as Argos (Pinciroli *et al.*, 2012) with communications simulators such as NS-3 (NS3 Consortium, 2016). The available simulators are listed in Section 2.5. We will detail the design of a software system that integrates multi-robot and wireless simulation, and then list challenges to its implementation.

## 8.1 Design of Integrated Simulator

As mentioned above, our design involves two components: a *multi robotic simulator (MRS)* and a *Wireless Communication Simulator* (WCS) that simulates a lossy network through which robots intercommunicate. The MRS simulates the kinematics and dynamics of the robot team; the WCS handles inter-robot communication. Wireless losses, such as path loss, propagation delay and multipath fading are simulated by the WCS. A typical WCS can model communications using Wi-fi, ZigBee, Bluetoooth, and Infra-Red. A robot wishing to be part of a distributed decision making process such as leader election sends a message to the WCS, requesting that it be included in the election. Modules within the WCS arbitrate the robots' decisions, and the result of the decision is sent back to the team through the WCS. This design decouples the MRS from the WCS, making it possible for an arbitrary pair of simulators exchange messages using a socket pair.

### 8.1.1 Execution of Boundary Coverage using Algorithm 1

We will illustrate the execution of the DCS of Algorithm 1 using the MRS/WCS pair. Suppose we are simulating a robot team attaching to a load placed in a typical lab ennvironment. A configuration file containing data about initial robot poses, obstacles in the lab, the load boundary, and the transmitting and receiving characteristics of the Wi-fi adapater of each robot is provided as input to the integrated software. The MRS uses the data about robot poses and the obstacles to simulate the dynamics of robot motion. The WCS uses the data about obstacles and the Wi-fi adapters to decide which packets get transmitted, and which get dropped *en route* to their destination due to network losses.

Robots initially move randomly in the lab in an attempt to locate the boundary $\mathscr{B}$. Since no communication is involved at this stage, the MRS computes robot poses at each time instant using its dynamical models. Subsequently, the robot team assembles at the white end of $\mathscr{B}$, and its robots form a network, which is subsequently simulated by the WCS. The robot that gets elected to traverse $\mathscr{B}$ is also decided by an election process brokered by the WCS. Because of packet losses, this election process may be faulty, or may abort. In the absence of packet losses, Algorithm 1 successfully executes, and the simulation ends with robots forming a connected network across $\mathscr{B}$.

## 8.2 Current Challenges to Implementation

We encountered the following challenges to implementing the integrated simulator designed in Section 8.1. A publicly available RoboNetSim patch (Kudelski *et al.*, 2013)

that equips ARGoS with communication abilities fails to work with the latest version of ARGoS . Also, ROS does not support multiple robots except with Rocon, which has a significant installation time and learning curve. We were thus unable to get it working by the submission date for this dissertation. Similarly, due to numerous software issues, we have as yet been unable to get Pylayers to work in environments other than those furnished in its manual. The lack of a standardized solution to multirobot communication has thus prevented us from designing and implementing communication-based experiments. Further, working directly on hardware with the Pheeno robotic platform (Wilson *et al.*, 2016) developed in our lab proved time-consuming, for getting even a pair of Pheenos to communicate on an ad-hoc network required 2-3 days to setup.

### *8.2.1 Steps to get a Working Simulation*

We outline a set of options for a future researcher to get a working simulation of multi-robotic experiments with realistic Wi-fi communication. By far, the easiest would be to patch up ARGoS with NS-3, following the outline of the earlier RoboNetSim patch. We estimate that this effort would take about 1.5 months. A more sophisticated option would be to use Rocon in combination with NS-3, which would involve the creation of a new patch that integrates the two simulators. The main purpose of these simulations is to obtain an understanding of what failures to expect in a particular environment, and how well the application-layer protocols for BC scale with the number of robots. Subsequently, the researcher could test the same protocols using GENI, Pheeno robots, or similar hardware.

# Chapter 9

# CONCLUSION AND FUTURE WORK

**Sources:** (Kumar *et al.*, 2013; Pavlic *et al.*, 2013a, 2015; Wilson *et al.*, 2014; Kumar and Berman, 2014, 2016b,a)

We will now summarize our findings from each chapter, and thereby conclude our response to the main problem statement.

## 9.1 Part 1 : Ant-Based CT

In Chapter 4, we conducted an experimental investigation of CT of a rigid artificial load by *Aphaenogaster cockerelli* ants. A stochastic hybrid system (SHS) model was developed to describe the dynamics of the load and the behavioral transitions of the ants during transport. The model was fit to the empirical mean of the data, and the resulting best-fit parameters were presented as reduced-order metrics of collective transport.

In future work, we plan to further validate our model by fitting both first-order and second-order moments to statistics from experimental data. We will also investigate how the best-fit parameters vary from optimized parameters that minimize criteria including path variance, load transport time, and transport team size.

We also plan to expand the model to incorporate other features of collective transport by ants. Additional behavioral states as well as heterogeneity and stochasticity within states

will be included. The modeled behaviors will include teams of individuals pulling and lifting with different time-varying forces. An important future direction is to adjust the transition rates so that they depend on factors such as the load position, load velocity, ant force applied to the load, and the number of ants attached. Especially in the uncoordinated phase before smooth transport, it is likely that the probability of an ant attaching to the front or back of the load is strongly determined by the load's nascent motion as well as the number of ants gathered around it. State-dependent transition rates can capture this initial behavior and still allow for the smooth motion that is the focus of this paper. As the uncoordinated phase has less directionality than the smooth phase discussed here, it will require augmenting the model for two-dimensional load motion.

In general, we hope to catalyze bio-inspired research on multi-robot transport teams. SHS frameworks are utilized in robotics, and so they have potential to be substrates for trans-disciplinary knowledge transfer.

## 9.2 Part 2 : Multi-Boundary Coverage

We have extended our prior work to the synthesis of communication-free stochastic control policies for robotic swarms that establish and maintain desired task allocations which are robust to changes in the environment. Moreover, our simulation results validate that design for multiple task types can be decomposed into isolated simpler single-type problems whose solutions are independent of each other. Finally, we have explored how the system provides several degrees of freedom for optimization of convergence rate and task-allocation variance. More generally, we have used a simple boundary-allocation problem

to demonstrate how specially designed irreversible processes can be used to implement higher level reversible processes that are less sensitive to environmental variations than the reversible processes typically associated studied in traditional statistical mechanics.

In our stochastic control approach, robots continue to bind and unbind randomly even after the equilibrium allocation has been reached, which apparently wastes a significant amount of power. This task switching at equilibrium, and its attendant power consumption, is a property of most stochastic strategies that already exist in the literature (Correll and Martinoli, 2004; Berman *et al.*, 2009; Odhner and Asada, 2010). In principle, a deterministic strategy could achieve the same distributions of robots to boundaries (possibly at allocations above the 75% parking constant) and then trigger robots to switch out of the costly allocation mode.

However, as has been discussed elsewhere for a particular deterministic–stochastic comparison (Dantu *et al.*, 2012), the deterministic approach would likely require significantly more communication between robots and additional sensing and navigation capabilities. Our stochastic approach allows for simpler robots that actually require *less* power and hardware for communication and sensing. In fact, the weight and physical space that is not required for communication and sensing may be used for additional battery capabilities. Additionally, our stochastic approach is insensitive to the communication and navigational errors that would be a significant concern for a deterministic approach. Furthermore, the continued operation of the stochastic allocation process allows the system to detect newly added boundaries and allocate to them automatically. Consequently, although the stochastic strategy we employ may have some performance weaknesses, it compensates for them by

being less costly to implement, less sensitive to sensing and actuation errors, and adaptive to disturbances and dynamic changes in the environment.

In future work, we will repeat these investigations for more arbitrary shapes. We also plan to gain a better understanding of the relationship between actual allocation ratio and the mean space between robots. Leveraging the several degrees of freedom in the enzymatic swarms (i.e., binding and unbinding probabilities, robot geometry, avoidance distances, and motion primitives that shape encounter-rate ratios), we will explore the design of optimal robot control policies to achieve convergence to desired allocations subject to other constraints, such as ensuring minimal allocation variance or settling within a specified time. We will extend our control approach to other scenarios in which robotic swarms must allocate among both static and dynamically moving regions, such as surveillance and target-tracking applications, and we will investigate how this approach can simplify other stochastic strategies such as swarm self-assembly.

## 9.3 Parts 3 and 4: Maintaining a Wireless Network across a Boundary

In Chapter 6, we analyzed statistical properties associated with stochastic coverage of a planar boundary by a robotic swarm. We developed a geometric interpretation of the probability that a given robot configuration is saturated, and we used results from order statistics to compute the pdfs of robot positions and inter-robot distances. We then focused on the condition of saturation, determining the geometry of the set of saturated robot configurations. We derived the marginal pdfs of robot positions and inter-robot distances for saturated configurations with just enough robots to saturate the boundary. Scenarios with larger robot

populations proved intractable for analysis, motivating us to formulate algorithms that generate samples from the space of saturated configurations to obtain the statistics of interest. We extended our results from point robots to finite robots, and we validated our formulas and algorithms using Monte Carlo simulations.

Chapter 7 has borrowed tools from complexity theory, order statistics, and computational geometry to compute the pdfs of a variety of spatial distributions of robots, as well as compute the complexity of determining $p_{\mathsf{con}}$ for the resulting communication graph $\mathscr{G}$. Our current work thus ties together a number of disparate disciplines in the pursuit of answers to the complexity of problems underlying the connectivity of random one dimensional networks. There are a number of ways in which its scope could be expanded or refined upon:

1. *Planar and Spatial Coverage* : The prototypical model of random networks in $\mathbb{R}^2$ is the Gilbert disk graph (Haenggi, 2012), in which each node communicates within a circle of radius $d$. While much is known about the properties of the RGGs (Penrose, 2003; Haenggi, 2012), little has been said about the properties of random connected graphs. Extending our results to $\mathbb{R}^2$ and $\mathbb{R}^3$ would enable us to analyze a large number of coverage scenarios. Further, the phenomenon of *percolation*, whereby $p_{\mathsf{con}}$ abruptly increases at a critical value of $d/s$, becomes manifest only in these dimensions. Bringing percolation theoretic methods to our extant toolkit would offer a fresh perspective into coverage problems.

2. *Incorporating Radio Propagation Models*: While all our results assumed idealized transmission and reception within the threshold $d$, any wireless communication model

will suffer from effects such as *path loss*, *interference* and *outage* (Haenggi, 2012). At present, we have no idea of which effects are significant enough to invalidate our computation of $p_{\mathsf{con}}$. Evidently, incorporating all these effects into our connectivity-related computations will result in an unwiedly model. Hence, a future step would be to develop realistic yet easy to understand models that judiciously take wireless losses into account.

3. *Incorporating odometric and sensing noise*: Every realistic model of robot motion will have odometric and sensing noise. Due to odometric noise, a differentially driven robot that has been imparted a constant linear velocity will drift from a straight line trajectory and fall on a banana-shaped curve (Thrun *et al.*, 2005). We are thus impelled to ask how odometric noise impacts connectivity. Further, in the presence of sensing noise, localization is inexact; consequently, computing order statistics and slacks in the presence of noise is thus a nontrivial task.

## 9.4 Part 5: Simulating Boundary Coverage

Our answer to Part 5 is incomplete due a large number of challenges standing in the way of implementing a multi-robot boundary coverage algorithm integrated with wireless communication over a noisy channel. Chapter 8 proposed a design for combining a Multi-Robot simulator with a Wireless communications simulator, wherein the two simulators exchange messages through a socket pair. However, due to the large number of challenges standing in the way of implementing the integrated simulator, the entirety of Chapter 8 addresses future work.

# REFERENCES


Alterovitz, R., S. Koenig and M. Likhachev, "Robot planning in the real world: research challenges and opportunities", National Science Foundation (2014).

Amiot, N., M. Laaraiedh and B. Uguen, "Pylayers: An open source dynamic simulator for indoor propagation and localization", in "Communications Workshops (ICC), 2013 IEEE International Conference on", pp. 84–88 (2013).

Becker, A., G. Habibi, J. Werfel, M. Rubenstein and J. McLurkin, "Massive uniform manipulation: Controlling large populations of simple robots with a common input signal", in "IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems (IROS)", pp. 520–527 (IEEE, Tokyo, Japan, 2013).

Bekey, G., R. Ambrose, V. Kumar, A. Sanderson, B. Wilcox and Y. Zheng, "International assessment of research and development in robotics", World Technology Evaluation Center Panel Report (2006).

Berg, M. D., O. Cheong, M. V. Kreveld and M. Overmars, *Computational Geometry: Algorithms and Applications* (Springer-Verlag TELOS, Santa Clara, CA, USA, 2008), 3rd ed. edn.

Berman, M., J. S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci and I. Seskar, "GENI: A federated testbed for innovative network experiments", Computer Networks **61**, 5–23 (2014).

Berman, S., Á. Halász, M. A. Hsieh and V. Kumar, "Optimized stochastic policies for task allocation in swarms of robots", IEEE Trans. Robot. **25**, 4, 927–937 (2009).

Berman, S., Q. Lindsey, M. S. Sakar, V. Kumar and S. C. Pratt, "Experimental study and modeling of group retrieval in ants as an approach to collective transport in swarm robotic systems", Proc. IEEE **99**, 9, 1470–1481 (2011).

Billera, L. J. and A. Sarangarajan, "The combinatorics of permutation polytopes", et al., Formal power series and algebraic combinatorics. Séries formelles et combinatoire algébrique pp. 23–27 (1994).

Boyd, S. and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, New York, NY, USA, 2004).

Brambilla, M., E. Ferrante, M. Birattari and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective", Swarm Intelligence **7**, 1, 1–41 (2013).

Censi, A., "Scan matching in a probabilistic framework", in "Proceedings of IEEE International Conference on Robotics and Automation, 2006.", pp. 2291–2296 (2006).

Chen, J., M. Gauci and R. Groß, "A strategy for transporting tall objects with a swarm of miniature mobile robots", in "Proc. of the 2013 International Conference on Robotics and Automation (ICRA)", pp. 863–869 (Karlsruhe, Germany, 2013).

Chen, K., K. Xu, S. Winburn, H. Shen, K.-C. Wang and Z. Li, "Experimentation of a MANET routing algorithm on the GENI ORBIT testbed", (2012).

Chib, S. and E. Greenberg, "Understanding the Metropolis-Hastings algorithm", The American Statistician **49**, 4, 327–335 (1995).

Correll, N., "Parameter estimation and optimal control of swarm-robotic systems: a case study in distributed task allocation", in "Proceedings of the 2008 IEEE International Conference on Robotics and Automation", pp. 3302–3307 (Pasadena, CA, USA, 2008).

Correll, N. and A. Martinoli, "Modeling and optimization of a swarm-intelligent inspection system", in "Proceedings of the Seventh International Symposium on Distributed Autonomous Robotics Systems (DARS 2004)", pp. 369–378 (Toulouse, France, 2004).

Czaczkes, T. J. and F. L. W. Ratnieks, "Cooperative transport in ants (Hymenoptera: Formicidae) and elsewhere", Myrmecol. News **18**, 1–11 (2013).

Dantu, K., S. Berman, B. Kate and R. Nagpal, "A comparison of deterministic and stochastic approaches for allocating spatially dependent tasks in micro-aerial vehicle collectives", in "Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on", pp. 793–800 (IEEE, 2012).

David, H. and H. Nagaraja, *Order Statistics* (John Wiley and Sons, Hoboken, NJ, USA, 2003).

Derakhshandeh, Z., S. Dolev, R. Gmyr, A. W. Richa, C. Scheideler and T. Strothmann, "Brief announcement: amoebot–a new model for programmable matter", in "Proceedings of the 26th ACM symposium on Parallelism in algorithms and architectures", pp. 220–222 (ACM, 2014a).

Derakhshandeh, Z., R. Gmyr, A. W. Richa, C. Scheideler, T. Strothmann and S. Tzur-David, "Infinite object coating in the amoebot model", arXiv preprint arXiv:1411.2356 URL `http://sops.cs.upb.de/publications/DGR+14.pdf` (2014b).

Devroye, L., *Non-uniform Random Variate Generation* (Springer-Verlag New York Inc., 1986), URL `http://www.eirene.de/Devroye.pdf`.

Diller, E. and M. Sitti, "Micro-scale mobile robotics", Foundations and Trends in Robotics **2**, 143–259 (2013).

Dolev, S., R. Gmyr, A. W. Richa and C. Scheideler, "Ameba-inspired self-organizing particle systems", arXiv preprint arXiv:1307.4259 (2013).

Dorigo, M. and et al., "Swarmanoid: A novel concept for the study of heterogeneous robotic swarms", Robotics & Automation Magazine, IEEE **20**, 4, 60–71 (2013).

Dvoretzky, A. and H. Robbins, "On the parking problem", Publications of the Mathematical Institute of the Hungarian Academy of Sciences pp. 209 – 224 (1964).

Dyer, M. and A. Frieze, "Computing the volume of convex bodies: a case where randomness provably helps", Probabilistic Combinatorics and Its applications **44**, 123–170 (1991).

Dyer, M. E. and A. M. Frieze, "On the complexity of computing the volume of a polyhedron", SIAM J. Comput. **17**, 5, 967–974, URL http://dx.doi.org/10.1137/0217060 (1988).

Evans, W. C., G. Mermoud and A. Martinoli, "Comparing and modeling distributed control strategies for miniature self-assembling robots", in "Proceedings of the 2010 IEEE International Conference on Robotics and Automation", pp. 1438–1445 (Anchorage, AK, USA, 2010).

Franks, N. R., "Teams in social insects: group retrieval of prey by army ants (*Eciton burchelli*, Hymenoptera: Formicidae)", Behav. Ecol. Sociobiol. **18**, 6, 425–429 (1986).

Gazebo, "Gazebo: Robot simulation made easy", URL www.gazebosim.org (2016).

Georgia Institute of Technology, et.al., "A Roadmap for US Robotics:From the Internet to Robotics", URL http://tiny.cc/e2jk9x (2013).

Gerkey, B. P., "AMCL ROS package", URL http://wiki.ros.org/amcl (2015).

Gillespie, D., "A rigorous derivation of the chemical master equation", Physica A **188**, 1-3, 404–425 (1992).

Gillespie, D., "Stochastic simulation of chemical kinetics", Annu. Rev. Phys. Chem. **58**, 35–55 (2007).

Glueck, D. H., A. Karimpour-Fard, J. Mandel, L. Hunter and K. E. Muller, "Fast computation by block permanents of cumulative distribution functions of order statistics from several populations", Communications in Statistics-Theory and Methods **37**, 18, 2815–2824 (2008).

Grünbaum, B., *Convex Polytopes* (Springer-Verlag New York Inc., 2002), second edn.

Guizzo, E., "Three engineers, hundreds of robots, one warehouse", Spectrum, IEEE **45**, 7, 26–34 (2008).

Gurarie, E., *Models and analysis of animal movements: From individual tracks to mass dispersal*, Ph.D. thesis, University of Washington (2008).

Haenggi, M., *Stochastic Geometry for Wireless Networks* (Cambridge University Press, 2012).

Hauert, S., S. Berman, R. Nagpal and S. N. Bhatia, "A computational framework for identifying design guidelines to increase the penetration of targeted nanoparticles into tumors", Nano Today **8**, 6, 566–576 (2013).

Hespanha, J. P., "Polynomial stochastic hybrid systems (extended version)", Tech. rep., University of California, URL http://www.ece.ucsb.edu/~hespanha/published/tr_pshs_hscc05.pdf (2004).

Hespanha, J. P. and A. Singh, "Stochastic models for chemically reacting systems using polynomial stochastic hybrid systems", Int. J. Robust Nonlinear Control. **15**, 15, 669–689 (2005).

Hu, J., J. Lygeros and S. Sastry, "Towards a theory of stochastic hybrid systems", in "Hybrid Systems: Computation and Control Third International Workshop Proceedings", edited by N. A. Lynch and B. H. Krogh, vol. 1790 of *Lecture Notes in Computer Science*, pp. 160–173 (Pittsburgh, PA, 2000).

Hutchinson, J. M. C. and P. M. Waser, "Use, misuse and extensions of "ideal gas" models of animal encounter", Biol. Rev. **82**, 3, 335–359 (2007).

Internet Engineering Task Force, "Optimized link state routing: RFC 7181", `http://tools.ietf.org/html/rfc7181` (2014).

Ivan Dryanovski, William Morris, Andrea Censi, "Laser Scan Matcher", URL `http://wiki.ros.org/laser_scan_matcher` (2015).

Jacquet, P., P. Mühlethaler, T. Clausen, A. Laouiti, A. Qayyum and L. Viennot, "Optimized link state routing protocol for ad hoc networks", in "Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International", pp. 62–68 (IEEE, 2001).

Jamalipour, A. and Y. Ma, *Intermittently Connected Mobile Ad Hoc Networks: from Routing to Content Distribution* (Springer Science + Business Media, New York, NY, USA, 2011).

Jerrum, M., *Counting, Sampling, and Integrating: Algorithms and Complexity*, Lectures in Mathematics (Birkhauser Verlag, 2003).

Jerrum, M., A. Sinclair and E. Vigoda, "A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries", Journal of the ACM (JACM) **51**, 4, 671–697 (2004).

Khatib, O., K. Yokoi, K. Chang, D. Ruspini, R. Holmberg and A. Casal, "Coordination and decentralized cooperation of multiple mobile manipulators", Journal of Robotic Systems **13**, 11, 755–764, URL `http://dx.doi.org/10.1002/(SICI)1097-4563(199611)13:11<755::AID-ROB6>3.0.CO;2-U` (1996).

Kitts, C. A. and M. B. Egerstedt, "Design, control, and applications of real-world multirobot systems", (2008).

Klavins, E., S. Burden and N. Napp, "Optimal rules for programmed stochastic self-assembly", in "Proceedings of Robotics: Science and Systems", (Philadelphia, PA, USA, 2006).

Kohlbrecher, S. and J. Meyer, "Hector SLAM ROS Package", URL `http://wiki.ros.org/hector_slam` (2015).

Kohlbrecher, S., J. Meyer, O. von Stryk and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation", in "Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)", (IEEE, 2011).

Konur, S., C. Dixon and M. Fisher, "Analysing robot swarm behaviour via probabilistic model checking", Robotics and Autonomous Systems **60**, 2, 199–213 (2012).

Kube, C. R. and E. Bonabeau, "Cooperative transport by ants and robots", Robot. Auton. Syst. **30**, 1–2, 85–101 (2000).

Kudelski, M., L. M. Gambardella and G. A. Di Caro, "RoboNetSim: An integrated framework for multi-robot and network simulation", Robotics and Autonomous Systems **61**, 5, 483–496 (2013).

Kumar, G. P. and S. Berman, "Statistical analysis of stochastic multi-robot boundary coverage", in "Robotics and Automation (ICRA), 2014 IEEE International Conference on", pp. 74–81 (2014).

Kumar, G. P. and S. Berman, "Analysis of the communication graph generated by stochastic boundary coverage", In preparation for SIAM Journal of Computing URL `https://arxiv.org/abs/1604.01452` (2016a).

Kumar, G. P. and S. Berman, "Stochastic boundary coverage by robotic swarms: Algorithms, analysis, and applications", Under Review for IEEE Transactions on Robotics **-**, -, URL `http://tiny.cc/zenarx` (2016b).

Kumar, G. P., A. Buffin, T. P. Pavlic, S. C. Pratt and S. M. Berman, "A stochastic hybrid system model of collective transport in the desert ant Aphaenogaster cockerelli", in "Proc. 16th Int'l. Conf. on Hybrid Systems: Computation and Control", HSCC'13, pp. 119–124 (2013).

Kumar, V. and N. Michael, "Opportunities and challenges with autonomous micro aerial vehicles", Int'l. Journal of Robotics Research **31**, 11, 1279–1291 (2012).

Liu, W. and A. F. T. Winfield, "Modeling and optimization of adaptive foraging in swarm robotic systems", Int. J. Robot. Res. **29**, 14, 1743–1760 (2010).

Mataric, M. J., N. P. Koenig and D. Feil-Seifer, "Materials for enabling hands-on robotics and stem education.", in "AAAI spring symposium: Semantic scientific knowledge integration", pp. 99–102 (2007).

Mather, T. W. and M. A. Hsieh, "Distributed robot ensemble control for deployment to multiple sites", in "Proceedings of Robotics: Science and Systems", (Los Angeles, CA, USA, 2011).

MathWorks , Inc., "dfittool, open distribution fitting app", URL `http://tiny.cc/y2jk9x` (2014).

Matthey, L., S. Berman and V. Kumar, "Stochastic strategies for a swarm robotic assembly system", in "Proceedings of the 2009 IEEE International Conference on Robotics and Automation", pp. 1953–1958 (Kobe, Japan, 2009).

Mavroidis, C. and A. Ferreira, "Nanorobotics: Past, present, and future", in "Nanorobotics", edited by C. Mavroidis and A. Ferreira, pp. 3–27 (Springer New York, 2013).

McLurkin, J. and et al., "A robot system design for low-cost multi-robot manipulation", in "IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems (IROS)", pp. 912–918 (2014).

Mehta, A., J. DelPreto and D. Rus, "Integrated codesign of printable robots", J. Mechanisms Robotics **7**, 2, 021015 (2015).

Meijering, E., O. Dzyubachyk and I. Smal, *Methods for Cell and Particle Tracking*, vol. 504 of *Methods in Enzymology*, chap. 9, pp. 183–200 (Elsevier, 2012).

Montemayor, G. and J. T. Wen, "Decentralized collaborative load transport by multiple robots", in "Proc. of the 2005 International Conference on Robotics and Automation (ICRA)", pp. 372–377 (2005).

Muller, G., "The efficiency of random parking", `http://tiny.cc/7r8a7x` (2007).

Muncherino, A., C. Lavor, L. Liberti and N. Maculan, *Distance Geometry: Theory, Methods, and Applications* (Springer Science + Business Media, New York, NY, USA, 2013).

Napp, N., S. Burden and E. Klavins, "Setpoint regulation for stochastically interacting robots", in "Proceedings of Robotics: Science and Systems", (Seattle, WA, USA, 2009).

NS3 Consortium, "Ns3 discrete event network simulator", URL `www.nsam.org` (2016).

Odhner, L. U. and H. Asada, "Stochastic recruitment control of large ensemble systems with limited feedback", J. Dyn. Syst., Meas., Control. **132**, 4 (2010).

O'Grady, R., C. Pinciroli, R. Groß, A. L. Christensen, F. Mondada, M. Bonani and M. Dorigo, "Swarm-bots to the rescue", in "Proceedings of the 10th European Conference on Artificial Life", vol. 5777 of *Lecture Notes in Computer Science*, pp. 165–172 (Springer-Verlag, Budapest, Hungary, 2009).

Open Source Robotics Foundation, "ROS: Robot Operating System", URL `www.ros.org` (2015).

Parker, L., "Collective manipulation and construction", in "Springer Handbook of Computational Intelligence", pp. 1395–1406 (Springer, 2015).

Pavlic, T. P., S. Wilson, G. P. Kumar and S. Berman, "An enzyme-inspired approach to stochastic allocation of robotic swarms around boundaries", in "Int'l. Symposium on Robotics Research (ISRR)", (Singapore, 2013a), URL `http://tiny.cc/sw88lx`.

Pavlic, T. P., S. Wilson, G. P. Kumar and S. Berman, "An enzyme-inspired approach to stochastic allocation of robotic swarms around boundaries", in "Proceedings of the 16th International Symposium on Robotics Research (ISRR 2013)", (Singapore, 2013b).

Pavlic, T. P., S. Wilson, G. P. Kumar and S. Berman, "Control of stochastic boundary coverage by multi-robot systems", ASME J. Dyn. Sys. Meas. Control **137**, 3, 034505, URL `http://dx.doi.org/10.1115/1.4028353` (2015).

Penrose, M., *Random Geometric Graphs*, Oxford Studies in Probability (Oxford University Press, 2003).

Pinciroli, C., V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, M. Birattari, L. M. Gambardella and M. Dorigo, "ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems", Swarm Intelligence **6**, 4, 271–295 (2012).

Renard, K., C. Peri and J. Clarke, "A performance and scalability evaluation of the ns-3 distributed scheduler", in "Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques", pp. 378–382 (ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2012).

Rényi, A., "On a one-dimensional problem concerning random space-filling", Publ. Math. Inst. Hung. Acad. Sci. **3**, 109–127 (1958).

Rubenstein, M., A. Cabrera, J. Werfel, G. Habibi, J. McLurkin and R. Nagpal, "Collective transport of complex objects by simple robots: Theory and experiments", in "Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems", AAMAS '13, pp. 47–54 (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2013), URL `http://dl.acm.org/citation.cfm?id=2484920.2484932`.

Rubenstein, M., A. Cornejo and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm", Science **345**, 6198, 795–799, URL `http://www.sciencemag.org/content/345/6198/795` (2014).

Schneider, C. A., W. S. Rasband and K. W. Eliceiri, "NIH Image to ImageJ: 25 years of image analysis", Nat. Methods **9**, 7, 671–675 (2012).

Sikiric, M. D. and Y. Itoh, *Random Sequential Packing of Cubes* (World Scientific Publishing Company, 2011).

Sinha, R., G. J. Kim, S. Niel and D. M. Shin, "Nanotechnology in cancer therapeutics: bioconjugated nanoparticles for drug delivery", Mol. Cancer Ther. **5**, URL `http://mct.aacrjournals.org/content/5/8/1909.long` (2006).

Stilwell, D. and J. Bay, "Toward the development of a material transport system using swarms of ant-like robots", in "Proceedings of the International Conference on Robotics and Automation (ICRA)", pp. 766–771 (1993).

Stonier, D., J. Lee and H. Kim, "Robotics in concert", URL `www.robotconcert.org` (2013).

Sugawara, K., D. Reishus and N. Correll, "Object transportation by granular convection using swarm robots", in "Distributed Autonomous Robotic Systems", STAR (Springer-Verlag, Baltimore, MD, 2012), URL `http://spot.colorado.edu/~dure7259/papers/Sugawara12.pdf`.

Sun, D. and J. Mills, "Manipulating rigid payloads with multiple robots using compliant grippers", Mechatronics, IEEE/ASME Transactions on **7**, 1, 23–34 (2002).

Talbot, J., G. Tarjus, P. Van Tassel and P. Viot, "From car parking to protein adsorption: an overview of sequential adsorption processes", Colloids and Surfaces A: Physicochemical and Engineering Aspects **165**, 1, 287–324 (2000).

Thrun, S., W. Burgard and D. Fox, *Probabilistic Robotics* (MIT press, 2005).

Tong, S., E. Fine, Y. Lin, T. J. Cradick and G. Bao, "Nanomedicine: Tiny particles and machines give huge gains", Annals of Biomedical Engineering pp. 1–17 (2013).

US Department of Defense, "Unmanned Systems Integrated Roadmap FY 2013-2038", URL `http://tiny.cc/ckpo9x` (2013).

Valiant, L. G., "The complexity of computing the permanent", Theoretical computer science **8**, 2, 189–201 (1979).

van Lint, J. and R. Wilson, *A Course in Combinatorics* (Cambridge University Press, New York, NY, USA, 2001).

Wang, C., G. Xie and M. Cao, "Controlling anonymous mobile agents with unidirectional locomotion to form formations on a circle", Automatica **50**, 4, 1100 – 1108, URL `http://dx.doi.org/10.1016/j.automatica.2014.02.036` (2014).

Westerman, G. and D. Bonnet, "Revamping your business through digital transformation", MIT Sloan Management Review **56**, 3, 10 (2015).

Wilensky, U., *NetLogo* (Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, USA, 1999), URL `http://ccl.northwestern.edu/netlogo/`.

Wilson, S., R. Gameros, M. Sheely, M. Lin, K. Dover, R. Gevorkyan, M. Haberland, A. L. Bertozzi and S. Berman, "Pheeno, a versatile swarm robotic research and education platform", IEEE Robotics and Automation Letters **PP**, 99, 1–1 (2016).

Wilson, S., T. P. Pavlic, G. P. Kumar, A. Buffin, S. C. Pratt and S. Berman, "Design of ant-inspired stochastic control policies for collective transport by robotic swarms", Swarm Intelligence **8**, 4, 303–327, URL `http://dx.doi.org/10.1007/s11721-014-0100-8` (2014).

Wood, R., R. Nagpal and G.-Y. Wei, "Flight of the Robobees", Scientific American **308**, 3, 60–65 (2013).

Zhang, G., G. K. Fricke and D. P. Garg, "Spill detection and perimeter surveillance via distributed swarming agents", Mechatronics, IEEE/ASME Transactions on **18**, 1, 121–129 (2013).

Zill, D., W. S. Wright and M. R. Cullen, *Advanced engineering mathematics* (Jones & Bartlett Learning, 2011).

# APPENDIX A

## LINKS TO VIDEO AND CODE RESOURCES

## A.1 Videos

The following URLs link to videos that are related to the dissertation.

1. Video of ants transporting a dime from the experiments in Section 4.2 :
   `https://goo.gl/ttnWeq`

2. NetLogo simulations of multi-boundary coverage in Chapter 5:
   `https://goo.gl/IzcWTt`

3. A robot traverses a load boundary, following it to an end-point as in Algorithm 1 :
   `https://goo.gl/llghzX`

4. Collective pushing of a heavy load in the ARGoS multi-robot simulator:
   `https://goo.gl/SBefNV`

## A.2 Code Repositories

The following URLs link to code repositories:

1. Numerical validation of several computations in Chapter 7 :
   `https://goo.gl/lbiy5l`
   `https://goo.gl/DCSMFg`

2. Boundary coverage with ARGoS in Chapter 8:
   `https://goo.gl/HcmxzE`

# APPENDIX B

## STATEMENT ON PERMISSION FROM CO-AUTHORS

Permission for including co-authored material in this dissertation has been obtained from my co-authors Prof. Spring Berman, Prof. Theodore Pavlic, Prof. Stephen Pratt, Sean Wilson, and Dr. Aurélie Buffin.